Networking protocols and administration

ITV8030

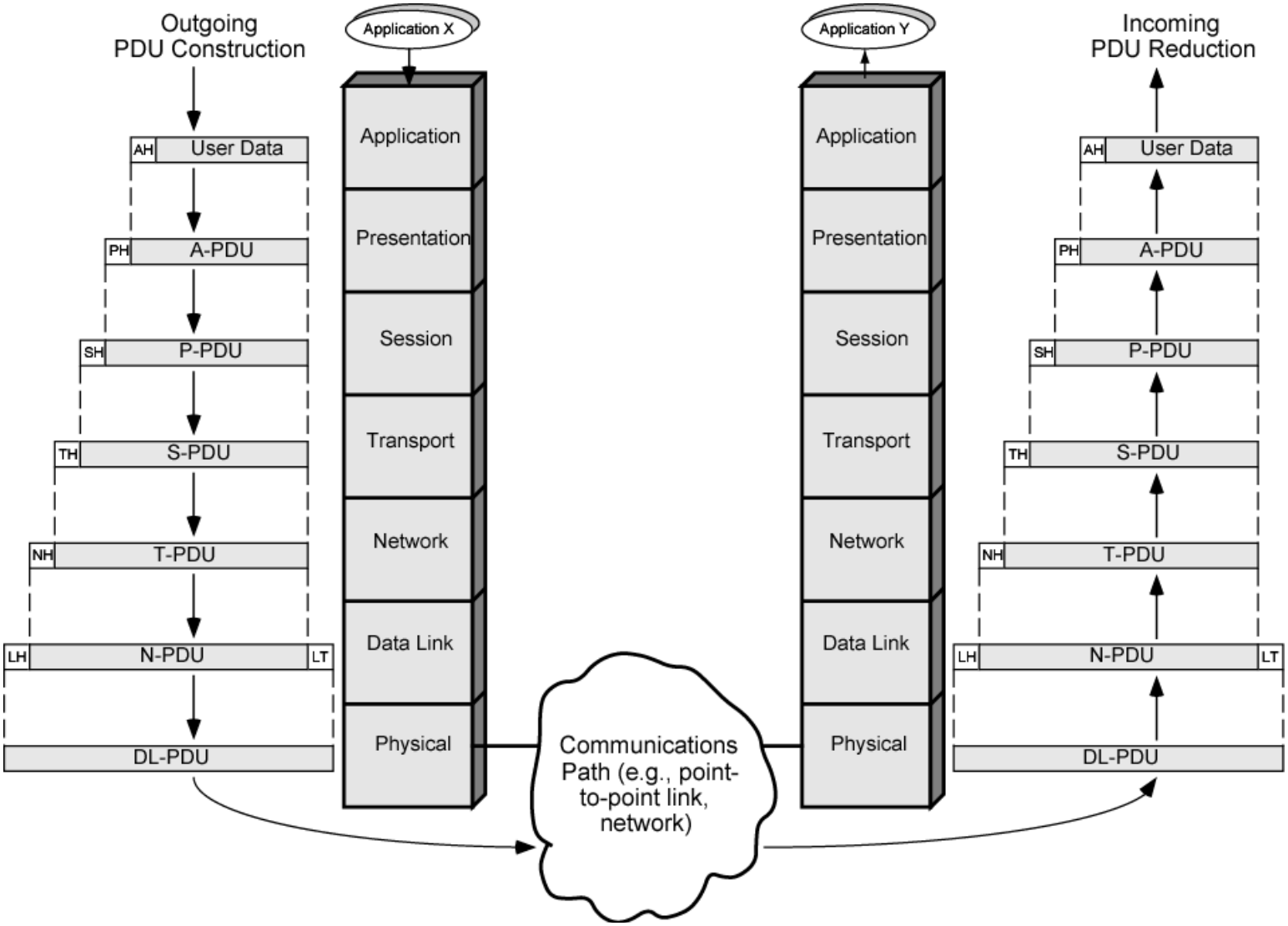
lecture 2: ethernet, ip packets

lecture plan

- Refresher
- How ethernet works
- Ethernet routing
- Ethernet frames (packets)
- ARP: find MAC addresses for IP-s
- IP packets

- Using mostly slides from Univ of Virginia / Univ of Toronto

OSI layers/headers



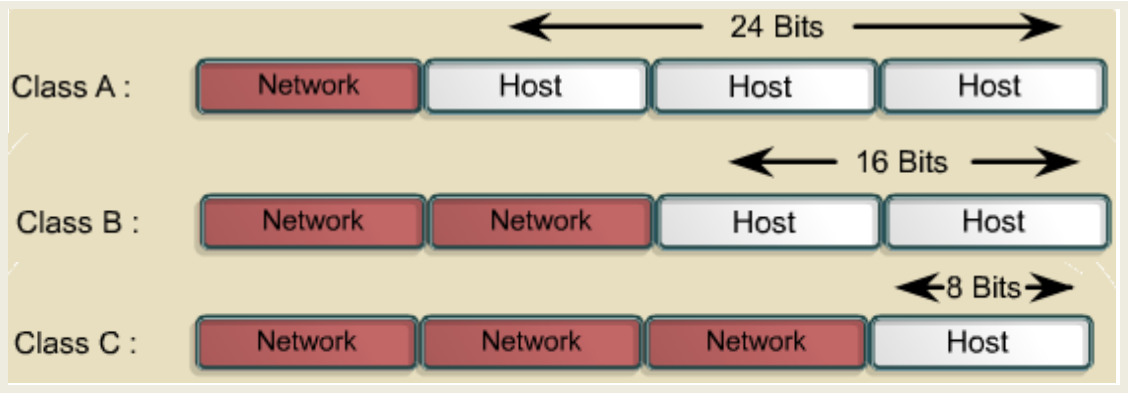
IP addresses contain two parts

- 4-byte IP address contains two parts:
  - Network address (routers know how to send packet to that network)
  - Host number (only local router knows how to send data inside local network)

- Three historic phases for hierachy organisation
  - Two fixed-length parts (very early): byte 1 (8 bits) for network, bytes 2-4 (24 bits) for host
  - Fixed types (A, B, C, D, E) of network addresses
  - Flexible selection of network address and local address parts (since 1993)

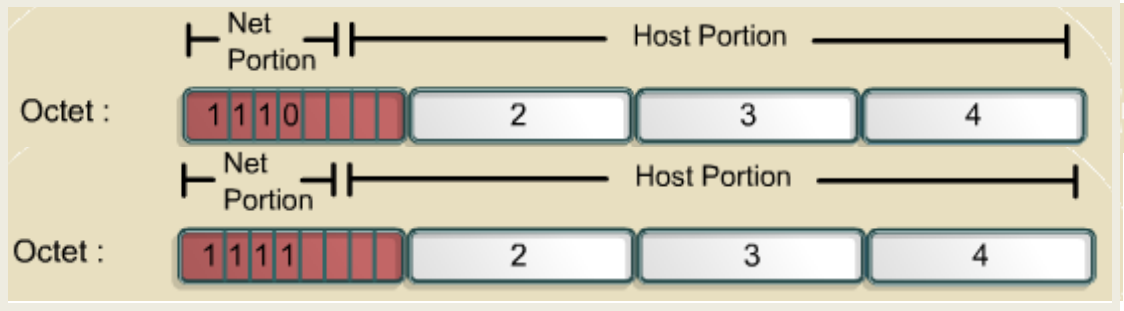
Class summary

IP addresses are divided into classes A,B and C to define large, medium, and small networks.



The Class D address class was created to enable multicasting.

IETF reserves Class E addresses for its own research.



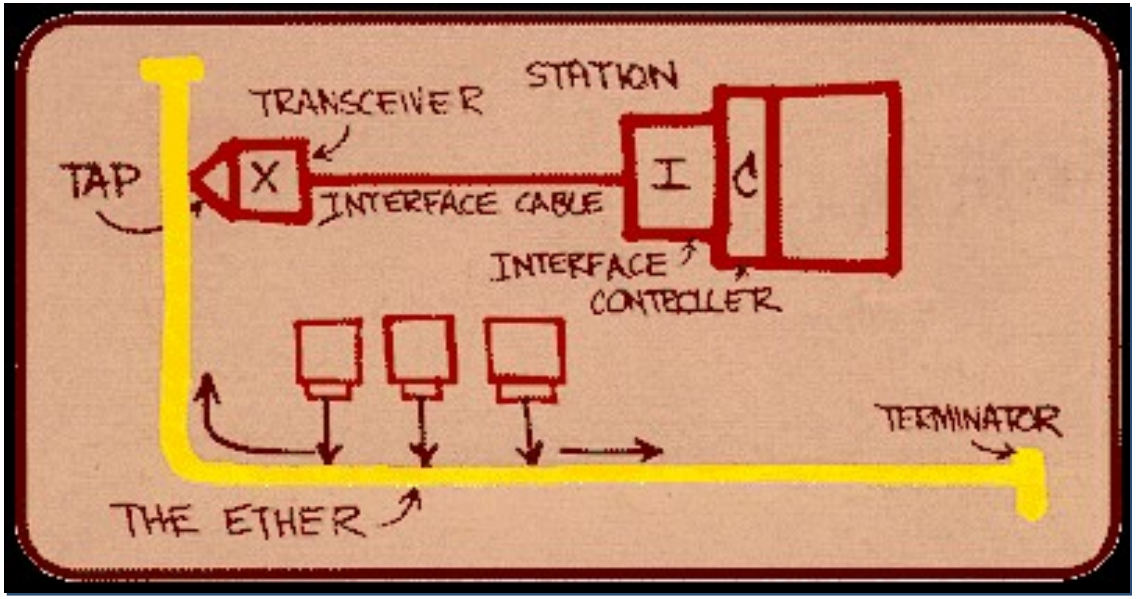
Address Class	High-Order Bits	First Octet Address Range	Number of Bits in the Network Address	Number of Networks	Number of Hosts per Network
Class A	0	0-127	8	126	16,777,216
Class B	10	128-191	16	16,384	65,536
Class C	110	192-223	24	2,097,152	254
Class D	1110	224-239	28	N/A	N/A

Ethernet

- Basic principles and routing

Main idea

- Connect all local machines with a wire
- Every machine can read everything on the wire
- Put data on wires in small packets containing destination machine MAC address
- Machines with a different MAC address will ignore the packet



Picture by Metcalfe:

inventor of internet
founder of 3COM

MAC address

- Used in ethernet, wifi, bluetooth, ATM, ...
- Should be globally unique (but does not have to!)
- Each ethernet card has its own default MAC built in
- 6 bytes, split into:
  - first three: manufacturer ID
  - following three: card ID for this manufacturer
- See http://coffer.com/mac_find/

MAC address can be changed

- Under Linux:
  - /etc/init.d/networking stop
  - ifconfig eth0 hw ether 02:01:02:03:04:08
  - /etc/init.d/networking start
- Under windows (several ways):
  - in the Ethernet adapter's Properties menu, in the Advanced tab, as "MAC Address", "Locally Administered Address", "Ethernet Address", "Physical Address" or "Network Address".
  - pass over the System Registry Keys under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}. Here settings for each network interface can be found and changed.

Collisions

- How to avoid the situation that several machines send packets at the same time?
- Answer: if you detect that somebody is sending something already, stop sendind and wait for a brief random amount of time, then try again.
- Statistically works fine!

carrier sense multiple access with collision detection (CSMA/CD)

- Main procedure and collision procedure
- Main procedure
  - Frame ready for transmission.
  - Is medium idle? If not, wait until it becomes ready and wait the interframe gap period (9.6 μs in 10 Mbit/s Ethernet).
  - Start transmitting.
  - Did a collision occur? If so, go to collision detected procedure.
  - Reset retransmission counters and end frame transmission.

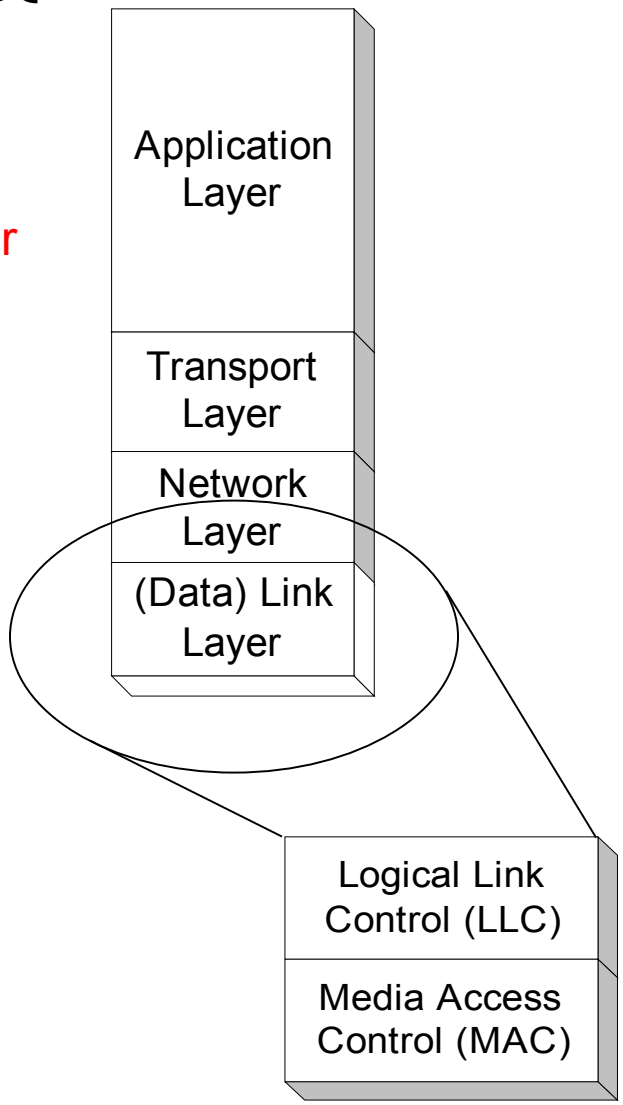
carrier sense multiple access with collision detection (CSMA/CD)

- Collision detected procedure
  - Continue transmission until minimum packet time is reached (jam signal) to ensure that all receivers detect the collision.
  - Increment retransmission counter.
  - Was the maximum number of transmission attempts reached? If so, abort transmission.
  - Calculate and wait random backoff period based on number of collisions.
  - Re-enter main procedure at stage 1.

TCP/IP Suite and OSI Reference Model

The TCP/IP protocol stack does not define the lower layers of a complete protocol stack

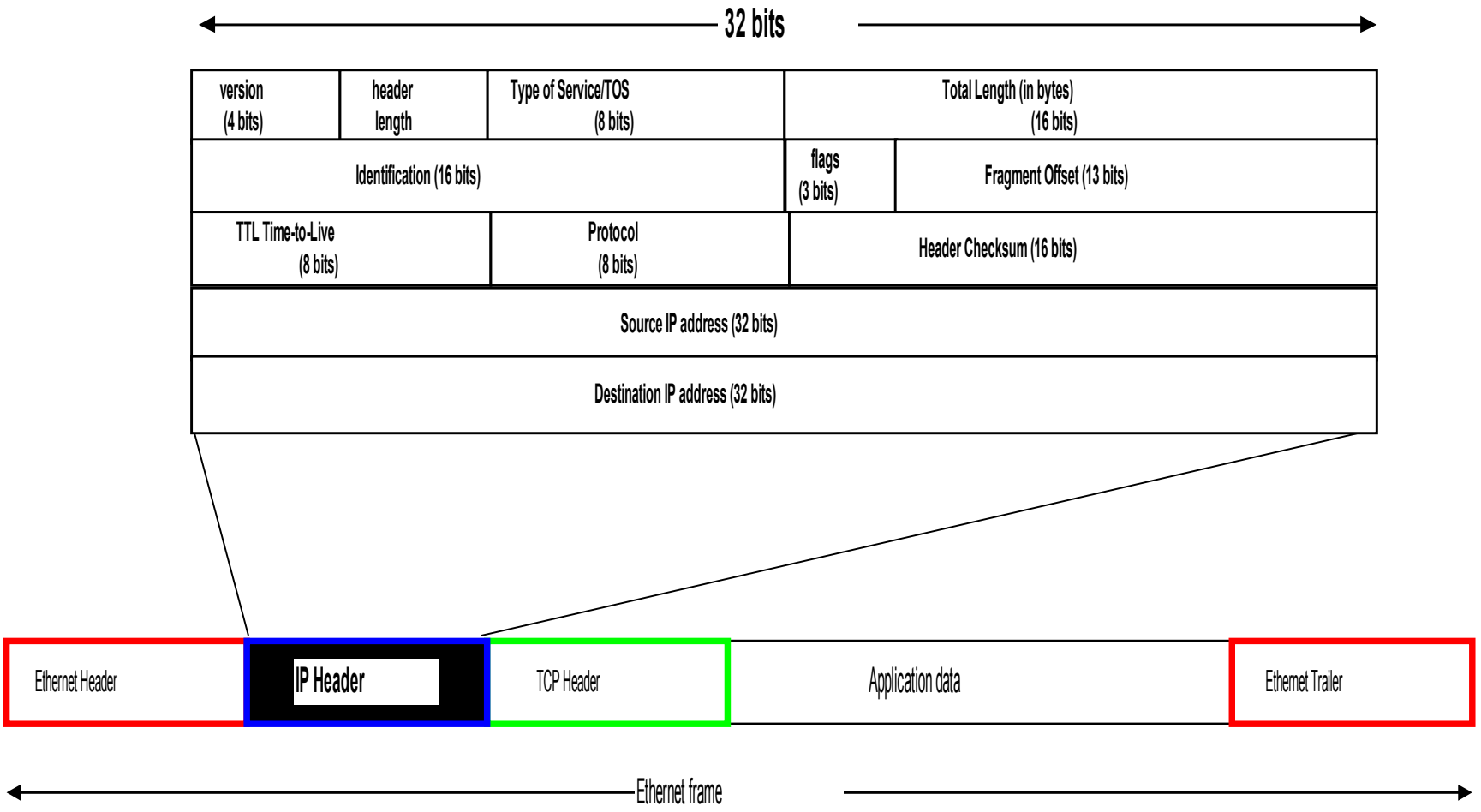
In this lecture, we will address how the TCP/IP protocol stacks interfaces with the data link layer



Data Link Layer

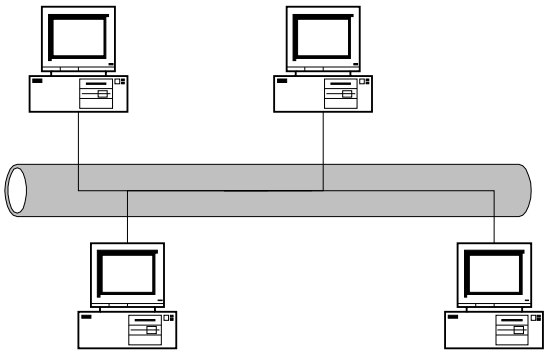
- The main tasks of the data link layer are:
  - Transfer data from the network layer of one machine to the network layer of another machine
  - Convert the raw bit stream of the physical layer into groups of bits ("frames")

IP packets in ethernet frames

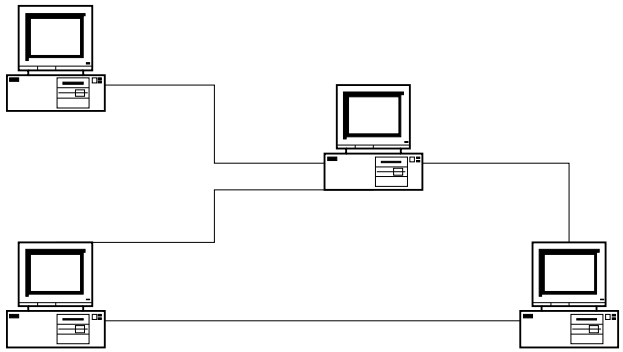


Two types of networks at the data link layer

- Broadcast Networks: All stations share a single communication channel
- Point-to-Point Networks: Pairs of hosts (or routers) are directly connected



Broadcast Network

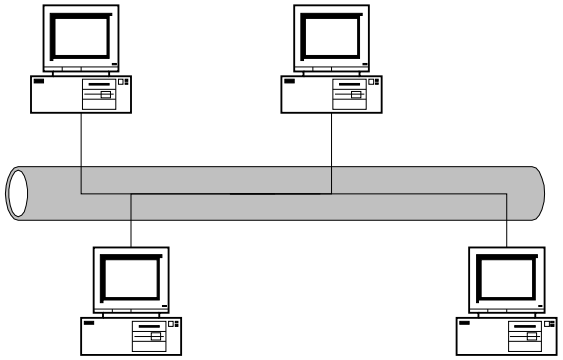


Point-to-Point Network

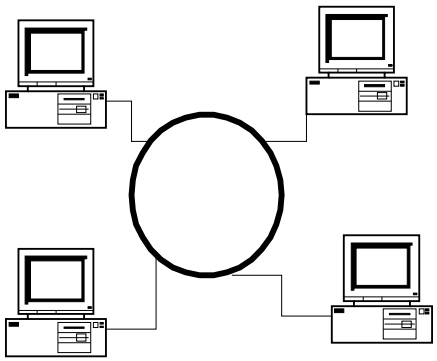
- Typically, local area networks (LANs) are broadcast and wide area networks (WANs) are point-to-point

Local Area Networks

- Local area networks (LANs) connect computers within a building or a enterprise network
- Almost all LANs are broadcast networks
- Typical topologies of LANs are **bus** or **ring** or **star**
- We will work with Ethernet LANs. Ethernet has a bus or star topology.



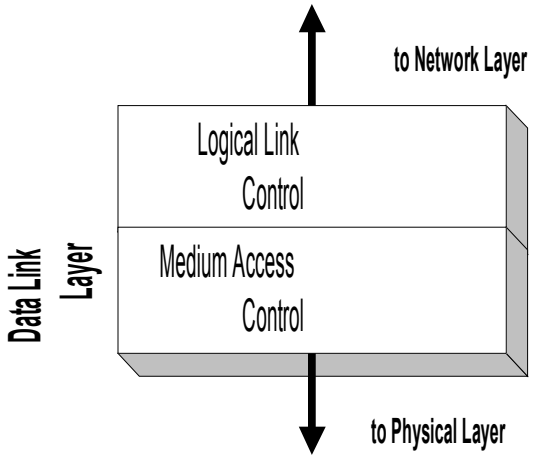
Bus LAN



Ring LAN

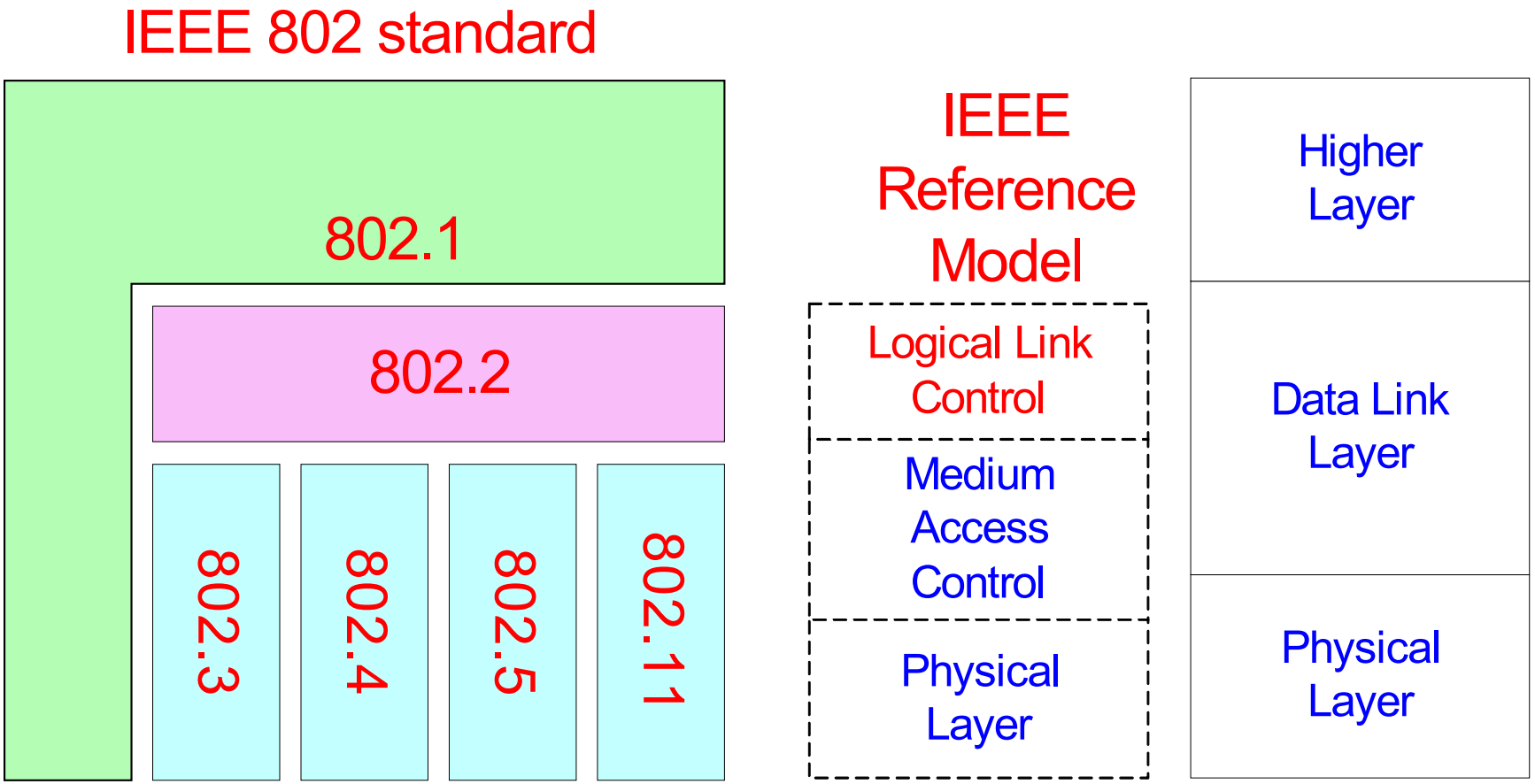
MAC and LLC

- In any broadcast network, the stations must ensure that only one station transmits at a time on the shared communication channel
- The protocol that determines who can transmit on a broadcast channel are called Medium Access Control (MAC) protocol
- The MAC protocol are implemented in the MAC sublayer which is the lower sublayer of the data link layer
- The higher portion of the data link layer is often called Logical Link Control (LLC)



IEEE 802 Standards

•IEEE 802 is a family of standards for LANs, which defines an LLC and several MAC sublayers



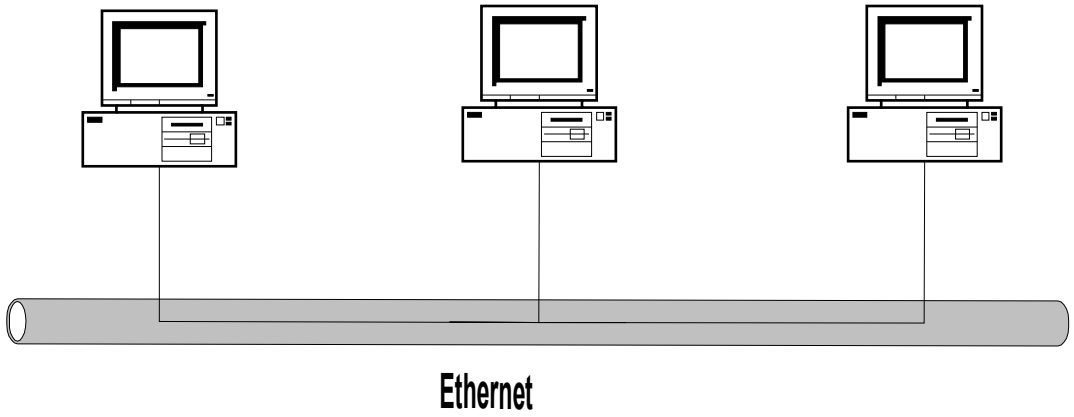
Different ethernet types

- Speed: 10Mbps -10 Gbps
- Standard: 802.3, Ethernet II (DIX)

- Most popular physical layers for Ethernet:
  - 10Base5 **Thick Ethernet:** 10 Mbps coax cable
  - 10Base2 **Thin Ethernet:** 10 Mbps coax cable
  - 10Base-T 10 Mbps Twisted Pair
  - 100Base-TX 100 Mbps over Category 5 twisted pair
  - 100Base-FX 100 Mbps over Fiber Optics
  - 1000Base-FX 1Gbps over Fiber Optics
  - 10000Base-FX 1Gbps over Fiber Optics (for wide area links)

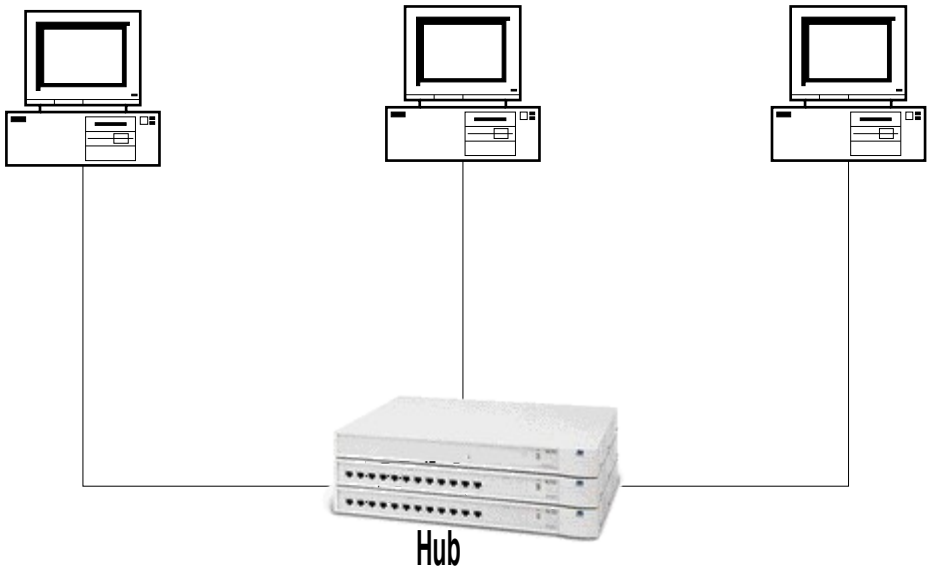
Old Bus Topology

- 10Base5 and 10Base2 Ethernets has a bus topology



Current Star Topology

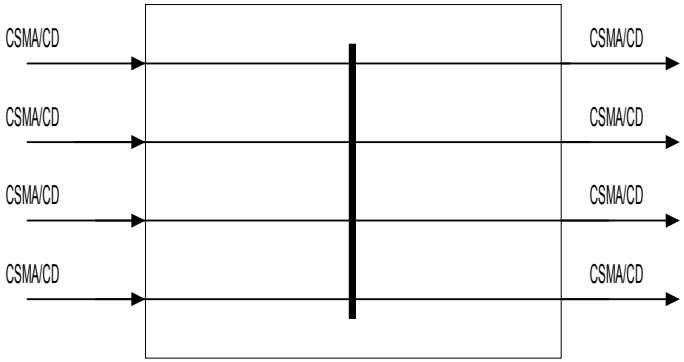
- Starting with 10Base-T, stations are connected to a hub in a star configuration



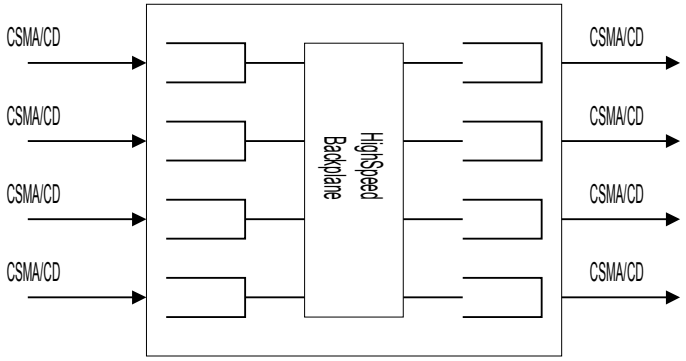
Ethernet Hubs vs. Ethernet Switches

- An **Ethernet switch** is a packet switch for Ethernet frames
  - Buffering of frames prevents collisions.
  - Each port is isolated and builds its own collision domain
- An **Ethernet Hub** does not perform buffering:
  - Collisions occur if two frames arrive at the same time.

Hub



Switch

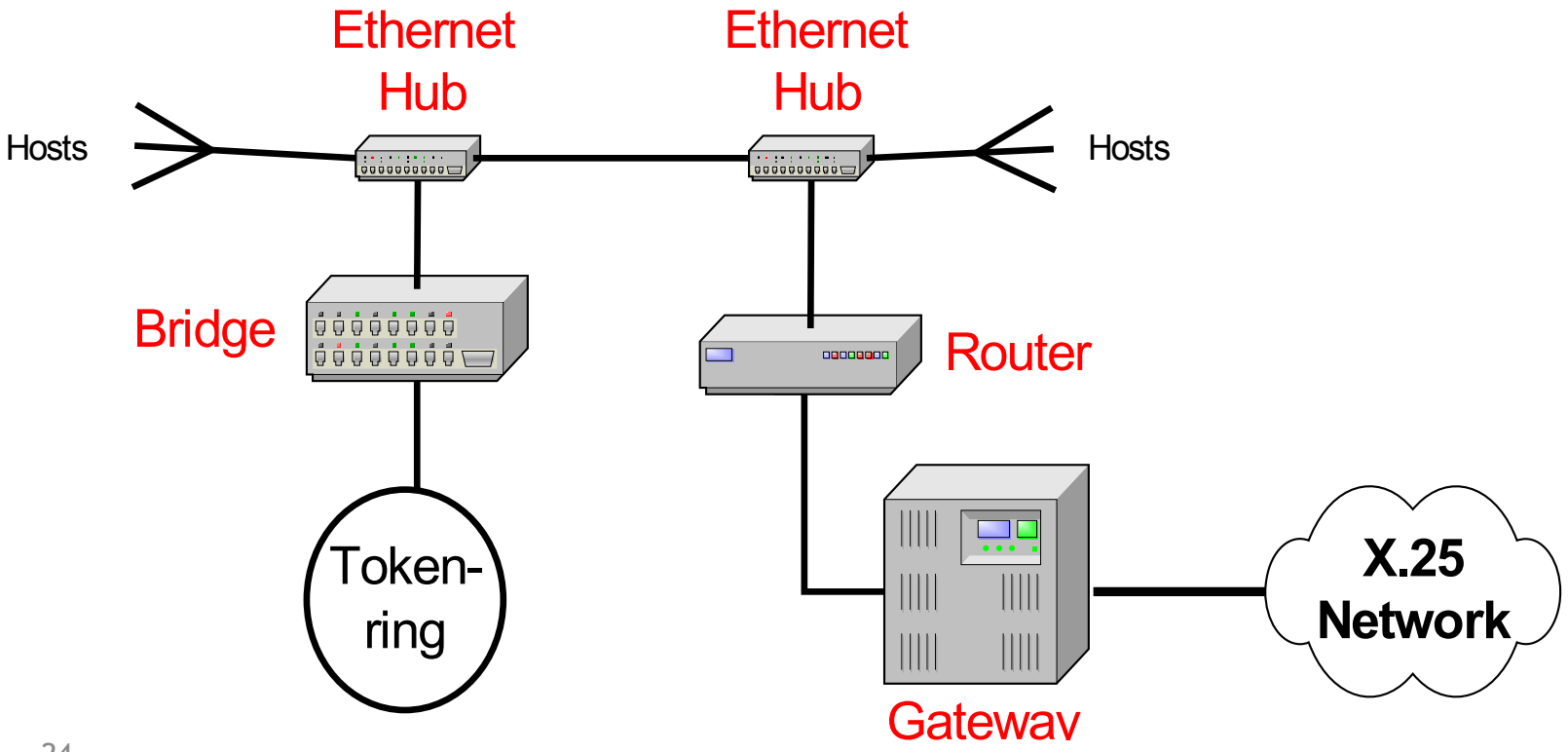


Input Buffers

Output Buffers

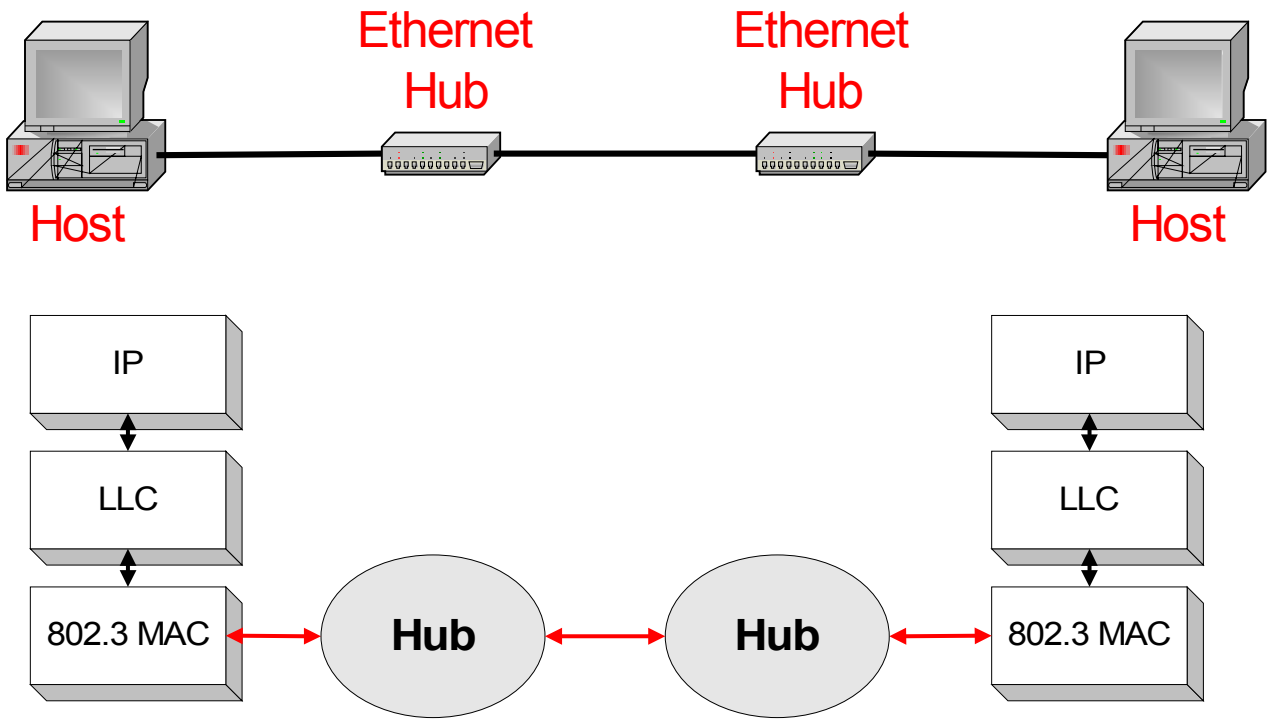
Introduction

- There are many different devices for interconnecting networks



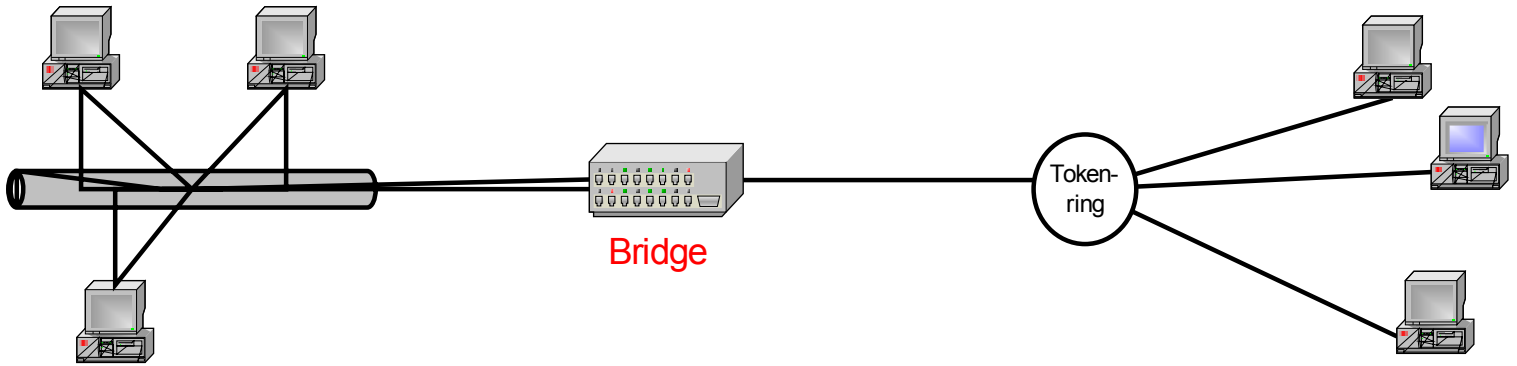
Ethernet Hub

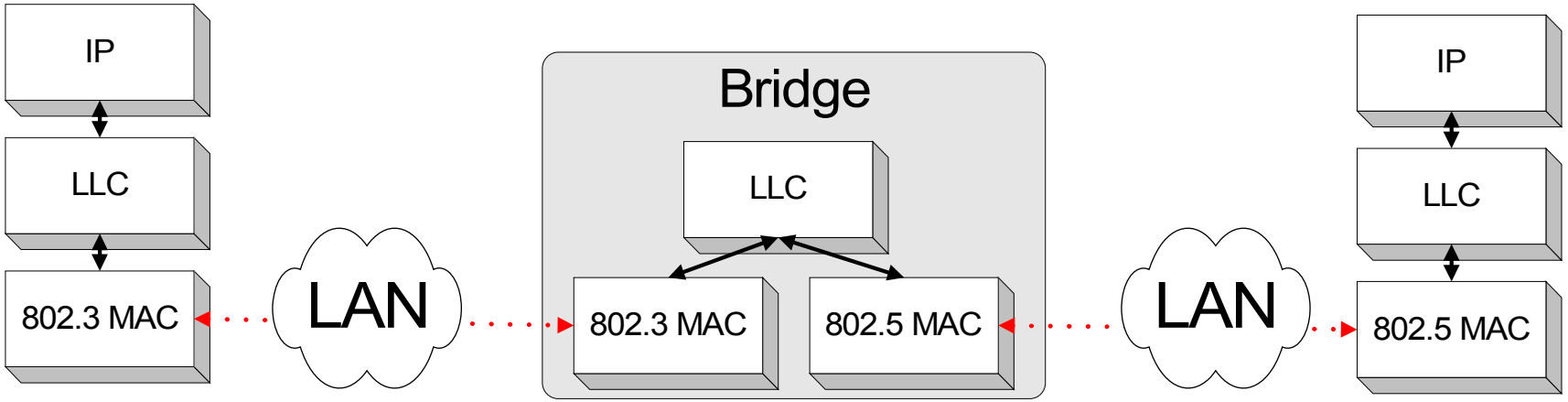
- Used to connect hosts to Ethernet LAN and to connect multiple Ethernet LANs
- Collisions are propagated



Bridges/LAN switches

- A *bridge or LAN switch* is a device that interconnects two or more *Local Area Networks (LANs)* and forwards packets between these networks.
- Bridges/*LAN switches* operate at the Data Link Layer (Layer 2)





Terminology: Bridge, LAN switch, Ethernet switch

There are different terms to refer to a data-link layer interconnection device:

- The term **bridge** was coined in the early 1980s.
- Today, the terms **LAN switch** or (in the context of Ethernet) **Ethernet switch** are used.

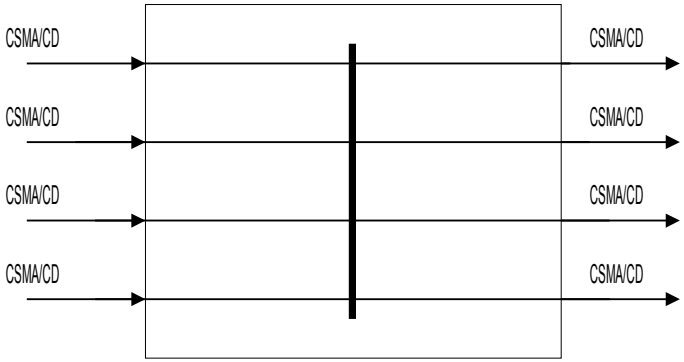
Convention:

- Since many of the concepts, configuration commands, and protocols for LAN switches were developed in the 1980s, and commonly use the old term *`bridge', we* will, with few exceptions, refer to LAN switches as bridges.

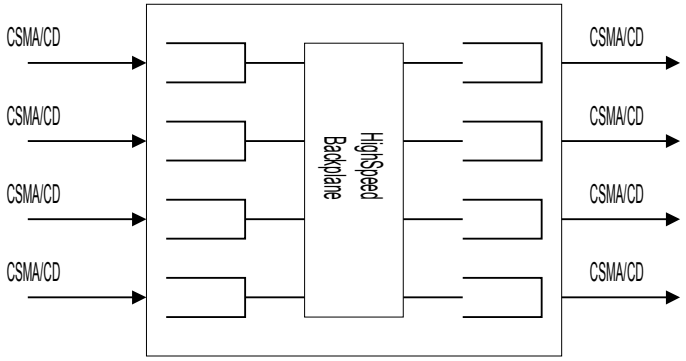
Ethernet Hubs vs. Ethernet Switches

- An **Ethernet switch** is a packet switch for Ethernet frames
  - Buffering of frames prevents collisions.
  - Each port is isolated and builds its own collision domain
- An **Ethernet Hub** does not perform buffering:
  - Collisions occur if two frames arrive at the same time.

Hub



Switch

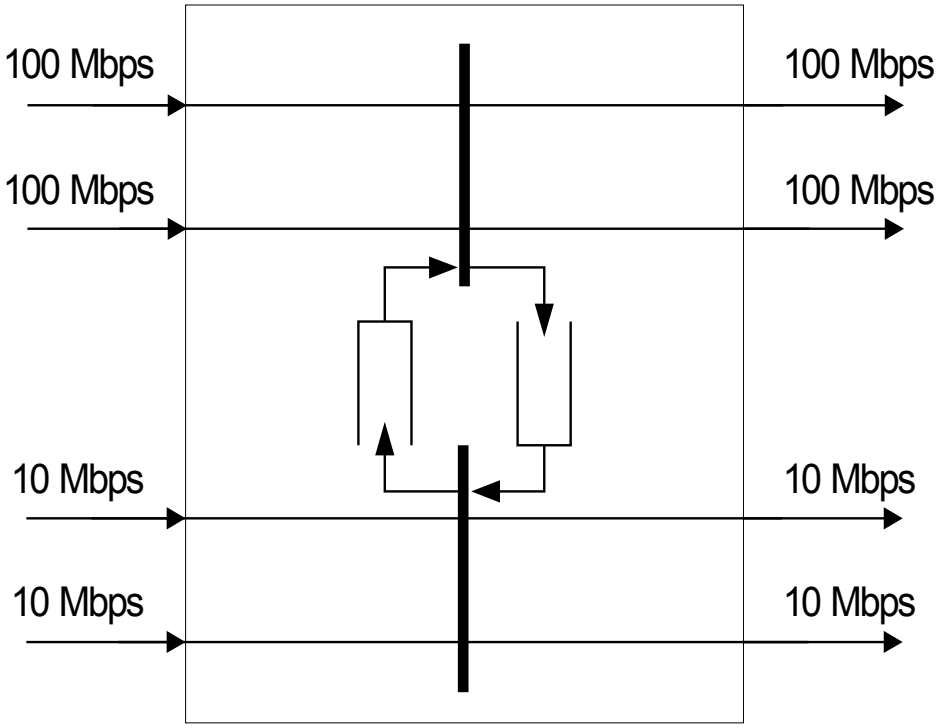


Input Buffers

Output Buffers

Dual Speed Ethernet hub

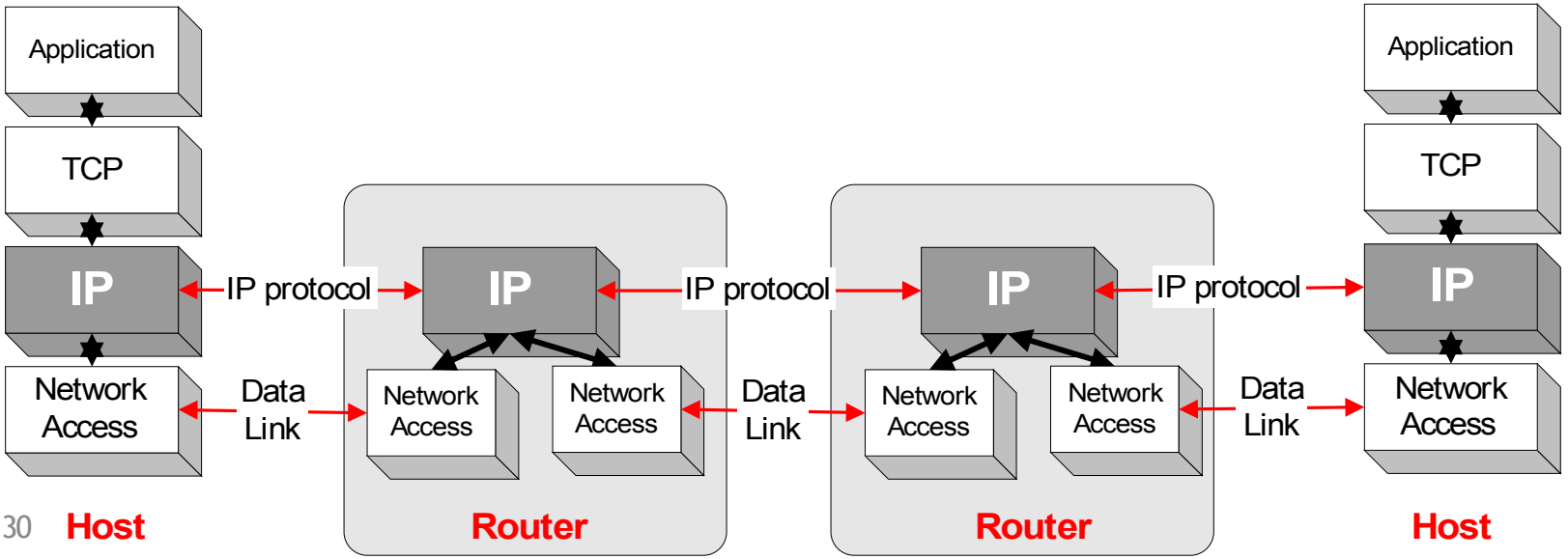
- Dual-speed hubs operate at 10 Mbps and 100 Mbps per second
- Conceptually these hubs operate like two Ethernet hubs separated by a bridge



Dual-Speed Ethernet Hub

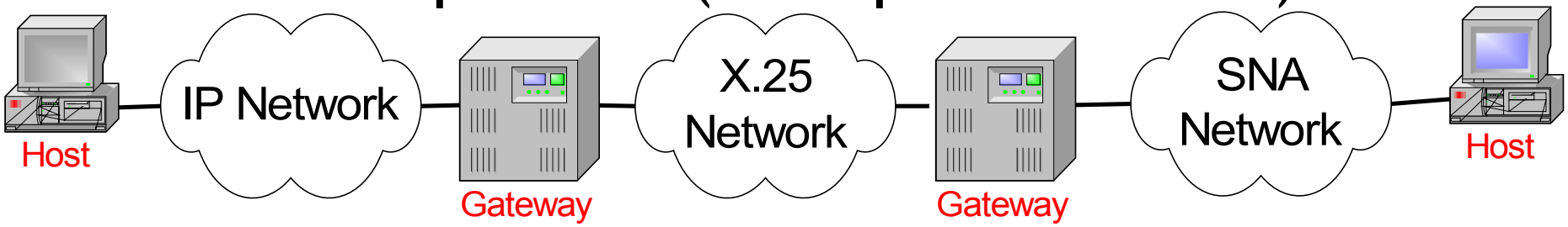
Routers

- Routers operate at the Network Layer (Layer 3)
- Interconnect IP networks



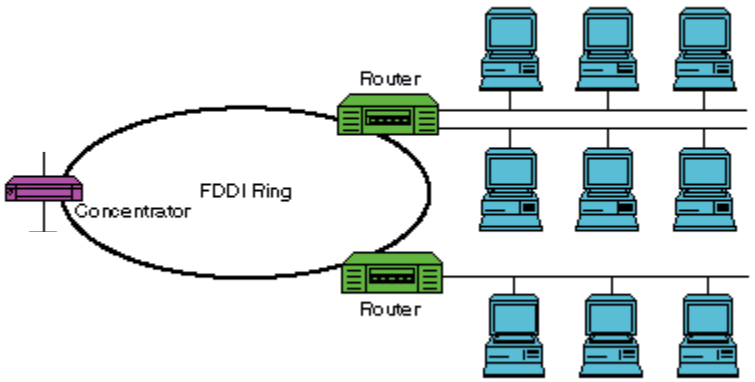
Gateways

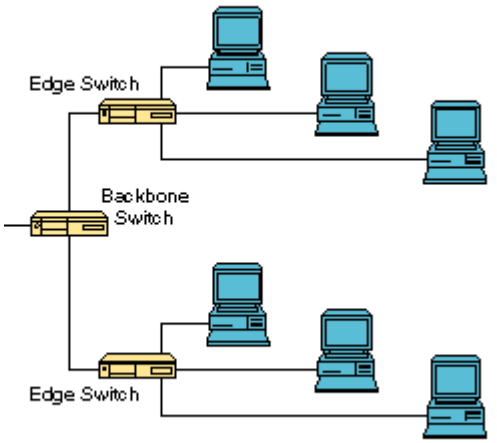
- The term "Gateway" is used with different meanings in different contexts
- "Gateway" is a generic term for routers (Level 3)
- "Gateway" is also used for a device that interconnects different Layer 3 networks and which performs translation of protocols ("Multi-protocol router")



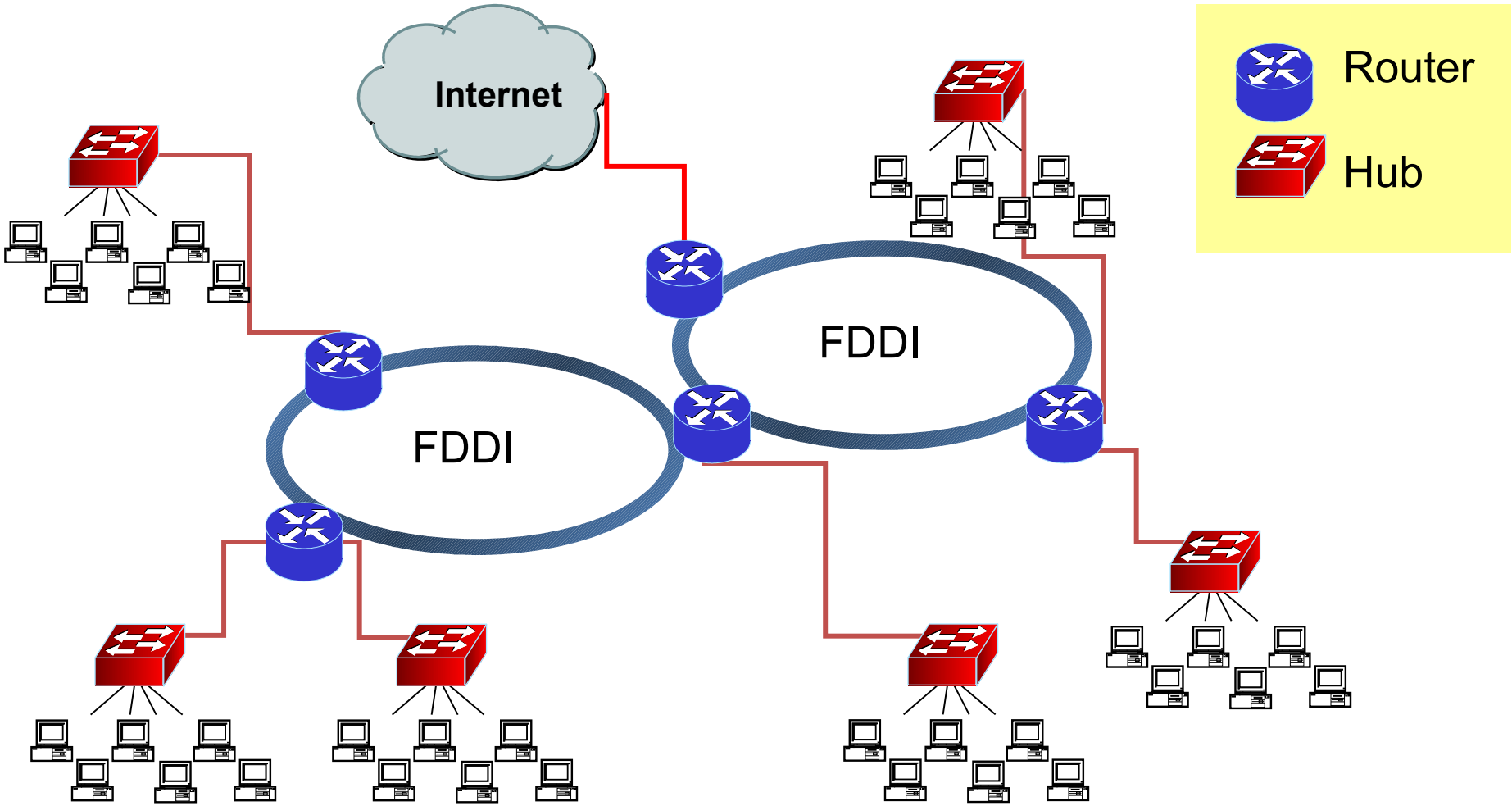
Bridges versus Routers

- An enterprise network (e.g., university network) with a large number of local area networks (LANs) can use routers or bridges
  - 1980s: LANs interconnection via bridges
  - Late 1980s and early 1990s: increasingly use of routers
  - LAN switches

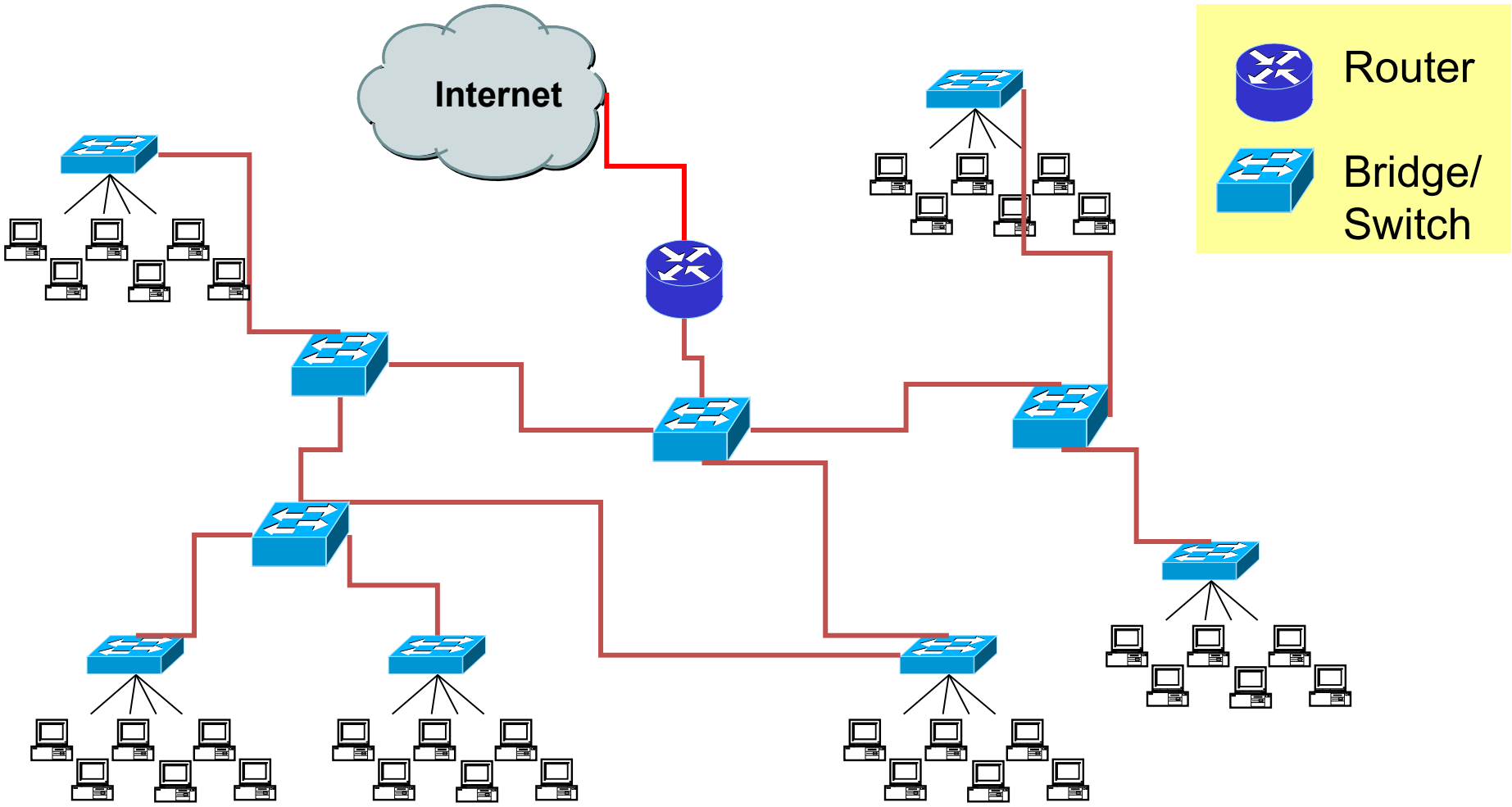




A Routed Enterprise Network

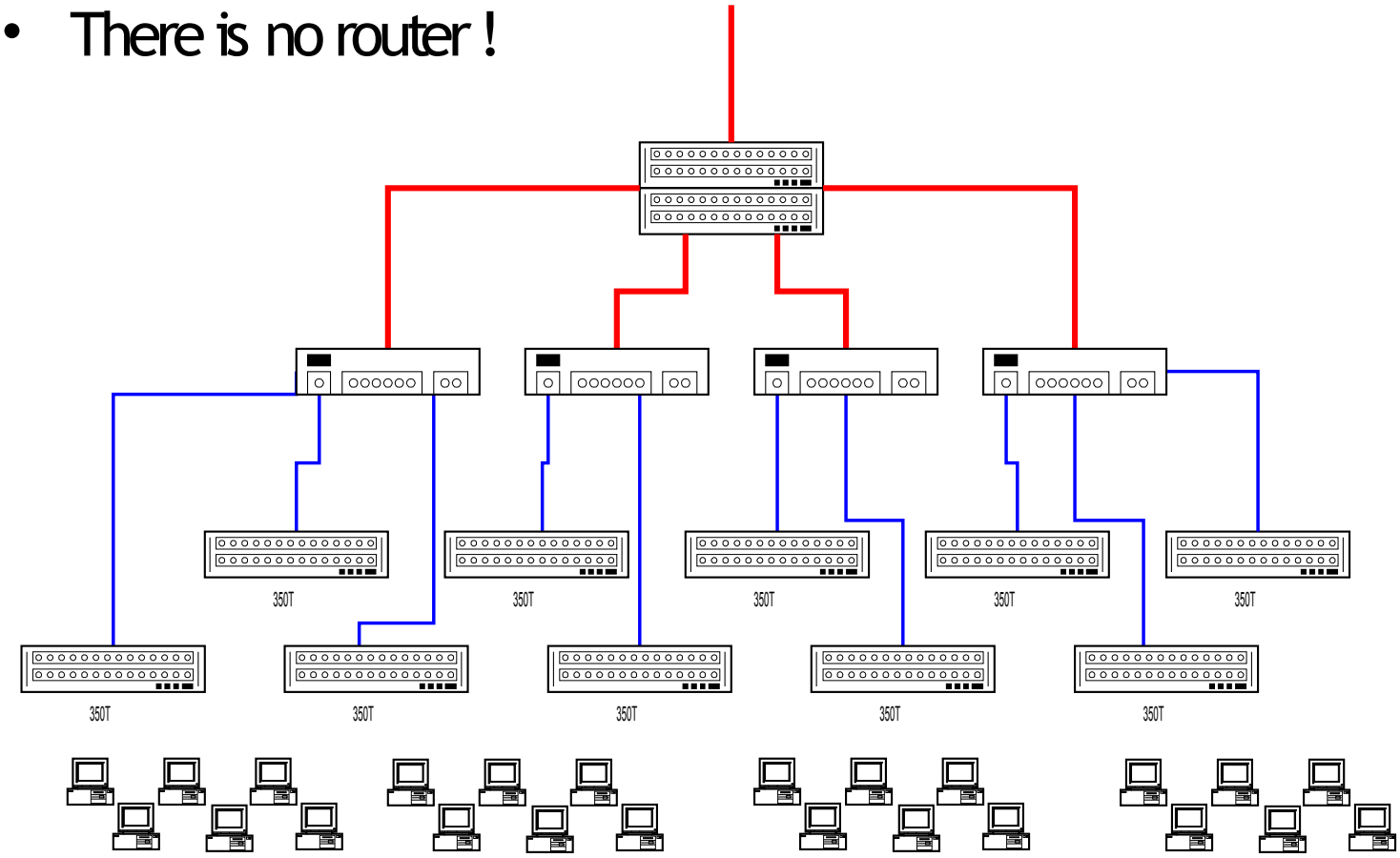


A Switched Enterprise Network



Example: Univ. of Virginia CS Department Network

- Design of the network architecture (Spring 2000)
- There is no router !



Interconnecting networks: Bridges versus Routers

Routers

- Each host's IP address must be configured
- If network is reconfigured, IP addresses may need to be reassigned
- Routing done via RIP or OSPF
- Each router manipulates packet header (e.g., reduces TTL field)

Bridges/LAN switches

- MAC addresses of hosts are hardwired
- No network configuration needed
- Routing done by
  - learning bridge algorithm
  - spanning tree algorithm
- Bridges do not manipulate frames

Bridges

Overall design goal: **Complete transparency**

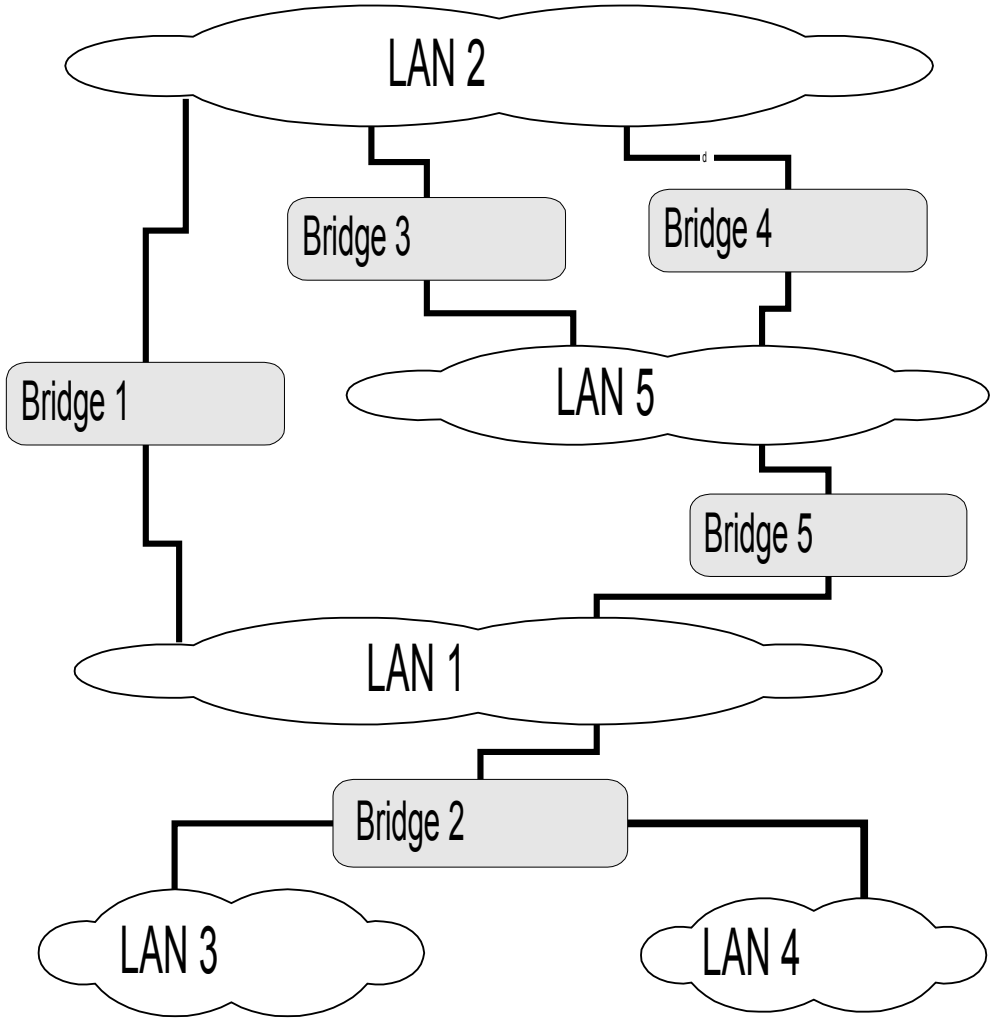
- "Plug-and-play"
- Self-configuring without hardware or software changes
- Bridges should not impact operation of existing LANs

Three parts to understanding bridges:

1. **Forwarding of Frames**
2. **Learning of Addresses**
3. **Spanning Tree Algorithm**

Need for a forwarding between networks

- What do bridges do if some LANs are reachable only in multiple hops ?
- What do bridges do if the path between two LANs is not unique ?



Transparent Bridges

- Three principal approaches can be found:
  - **Fixed Routing**
  - **Source Routing**
  - **Spanning Tree Routing** (IEEE 802.1d)

- We only discuss the last one in detail.

- Bridges that execute the spanning tree algorithm are called **transparent bridges**

(1) Frame Forwarding

- Each bridge maintains a **MAC forwarding table**
- Forwarding table plays the same role as the routing table of an IP router
- Entries have the form (MAC address, port, age), where

MAC address:	host name or group address
port:	port number of bridge
age:	aging time of entry (in seconds)

with interpretation:

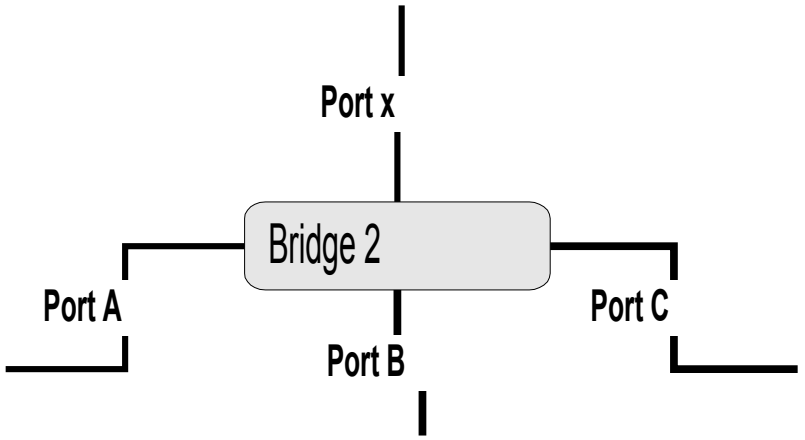
a machine with **MAC address** lies in direction of the **port** number from the bridge. The entry is **age** time units old.

MAC forwarding table →

MAC address	port	age
a0:e1:34:82:ca:34	1	10
45:6d:20:23:fe:2e	2	20

(1) Frame Forwarding

- Assume a MAC frame arrives on port x.



Is MAC address of destination in forwarding table for ports A, B, or C ?

Found?

Not found ?

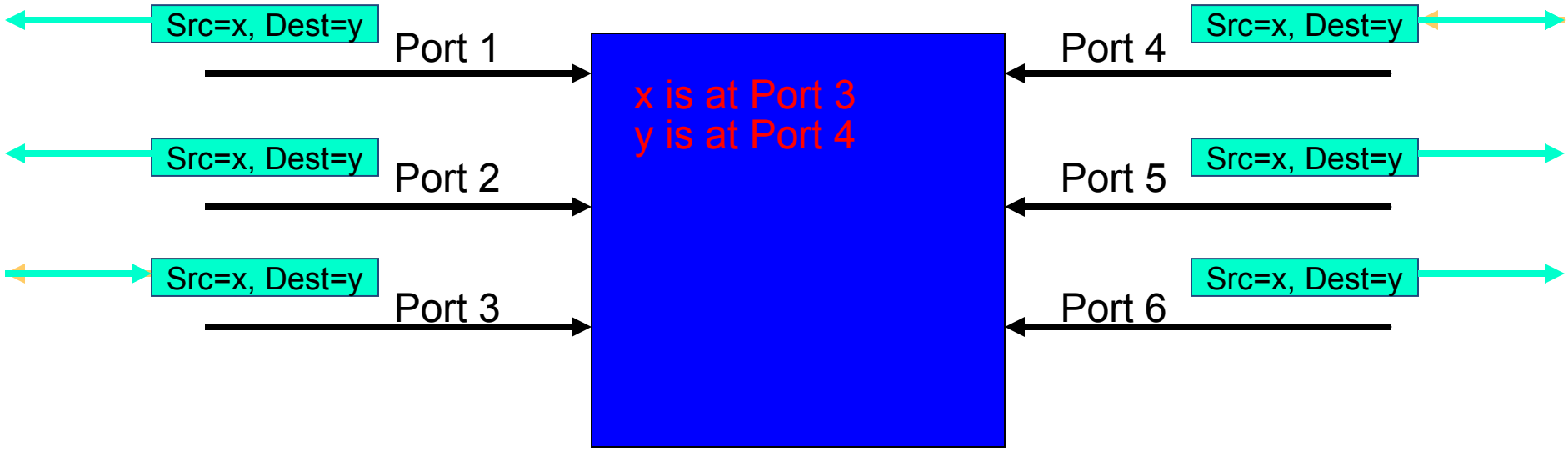
Forward the frame on the appropriate port

Flood the frame,
i.e.,
send the frame on all ports except port x.

(2) Address Learning (Learning Bridges)

- Routing tables entries are set automatically with a simple heuristic:

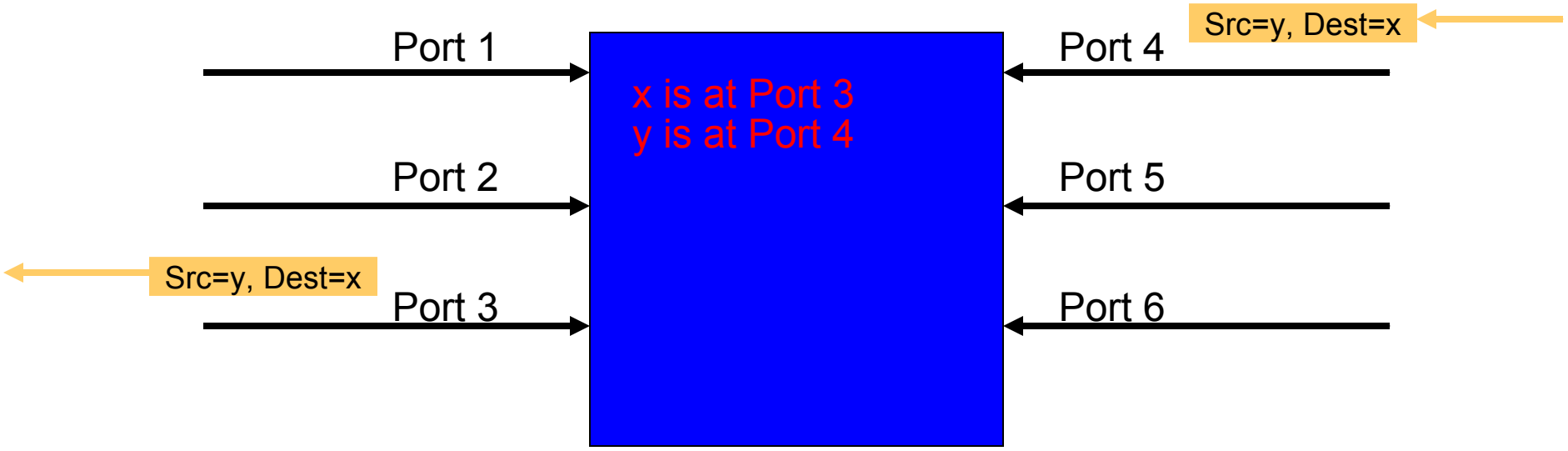
  The source field of a frame that arrives on a port tells which hosts are reachable from this port.



(2) Address Learning (Learning Bridges)

Learning Algorithm:

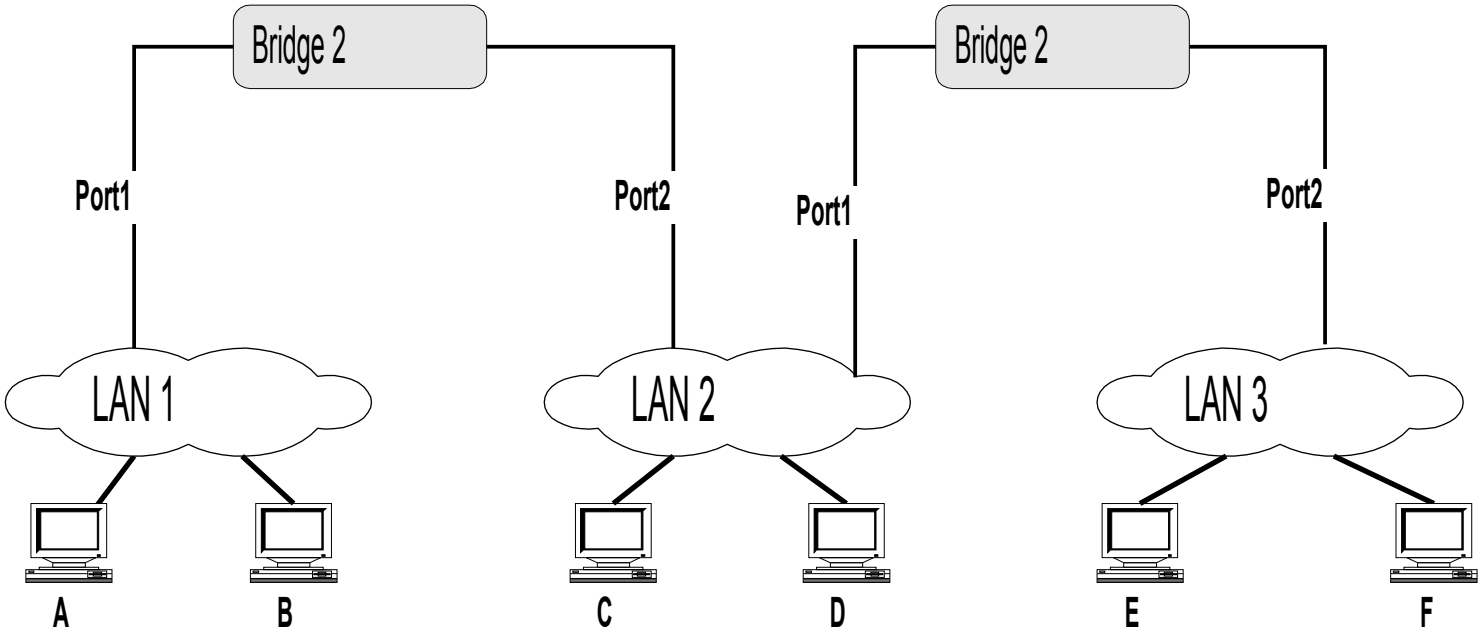
- For each frame received, the source stores the source field in the forwarding database together with the port where the frame was received.
- All entries are deleted after some time (default is 15 seconds).



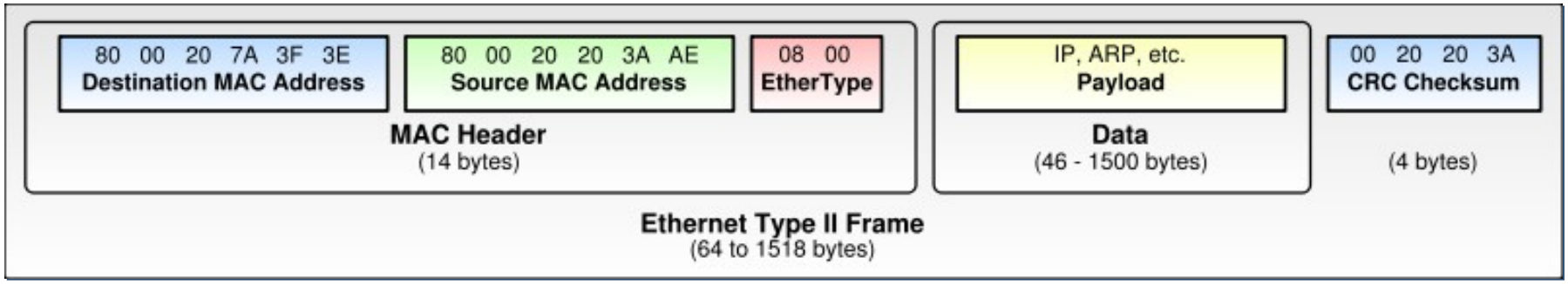
Example

Consider the following packets:
(Src=A, Dest=F), (Src=C, Dest=A), (Src=E, Dest=C)

What have the bridges learned?



Ethernet frame format



- Preamble 7 octets of 10101010
- Start-of-Frame-Delimiter 1 octet of 10101011

- **MAC destination 6 octets**
- **MAC source 6 octets**
- **Ethertype/Length 2 octets**
- **Payload 46-1500 octets**
- **CRC32 4 octets**

- (Postamble 1 octet of 01111110)
- Interframe gap 960 ns (100M)

Ethernet frame timing example

- MAC Preamble (+ SFD) 8 Bytes
- MAC Destination Address 6 Bytes
- MAC Source Address 6 Bytes
- MAC Type (or Length) 2 Bytes
- Payload (Network PDU) 46 Bytes
- Check Sequence (CRC) 4 Bytes
- Inter Frame Gap (9.6µs) 12 Bytes

- Total Frame Physical Size 84 Bytes

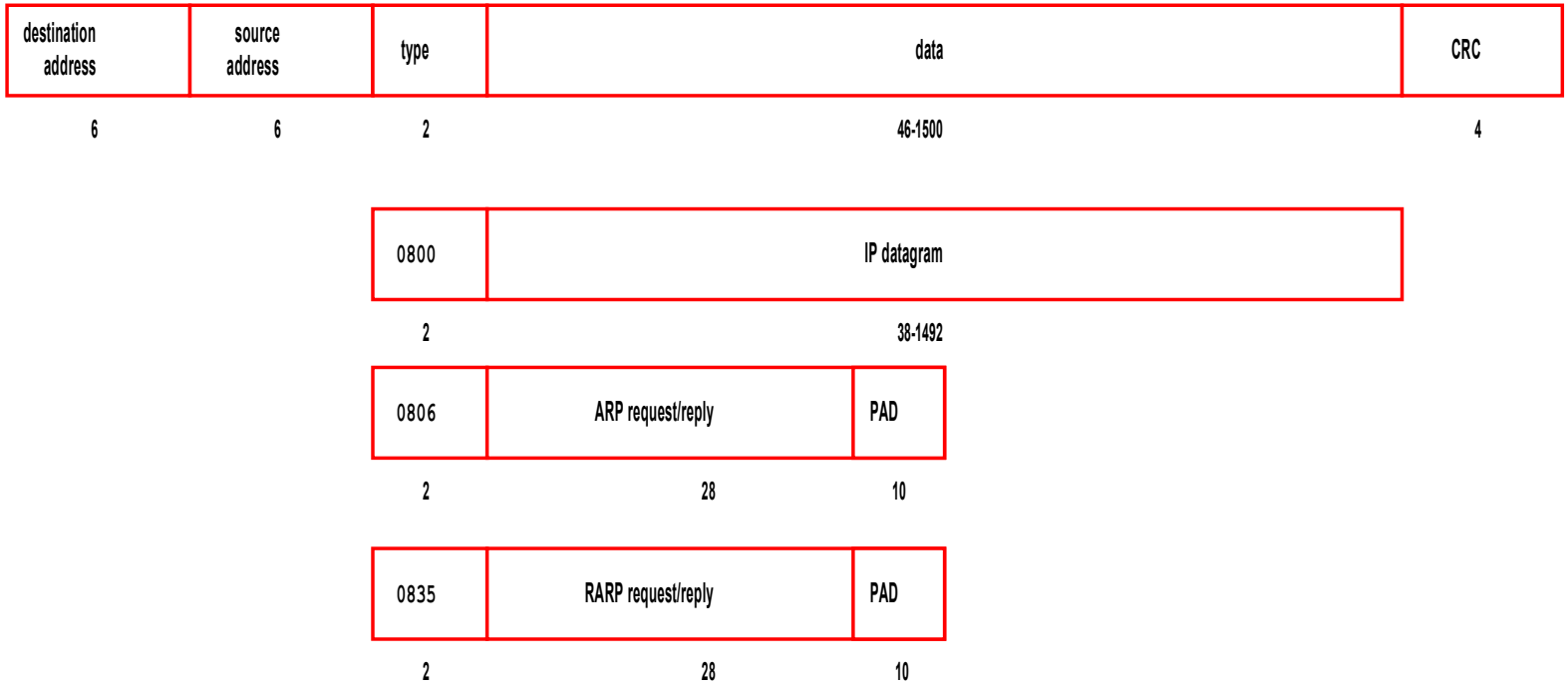
- The maximum number of frames per second for 46 byte payload is:
  - Ethernet Data Rate (bits per second) /Total Frame Physical Size (bits)
  - = 10 000 000 /(84 x 8)
  - = 14 880 frames per second.
- here payload transfer = 46*14800 = 680 800 bytes per second

Ethernet and IEEE 802.3: Any Difference?

- There are two types of Ethernet frames in use, with subtle differences:
- *"Ethernet" (Ethernet II, DIX)*
  - An industry standards from 1982 that is based on the first implementation of CSMA/CD by Xerox.
  - Predominant version of CSMA/CD in the US.
- **802.3:**
  - IEEE's version of CSMA/CD from 1985.
  - Interoperates with 802.2 (LLC) as higher layer.

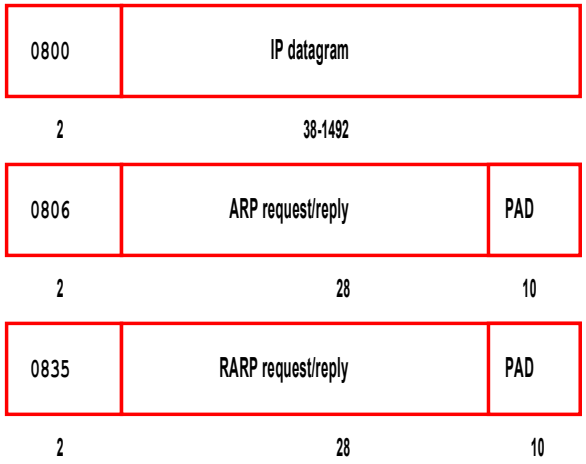
- **Difference for our purposes:** Ethernet and 802.3 use different methods to encapsulate an IP datagram.

Ethernet II, DIX Encapsulation (RFC 894)



IEEE 802.2/802.3 Encapsulation (RFC 1042)

- **destination address, source address:** MAC addresses are 48 bit
- **length:** frame length in number of bytes
- **DSAP, SSAP:** always set to `0xaa`
- **Ctrl:** set to 3
- **org code:** set to 0
- **type field:** identifies the content of the data field
- **CRC:** cylic redundancy check



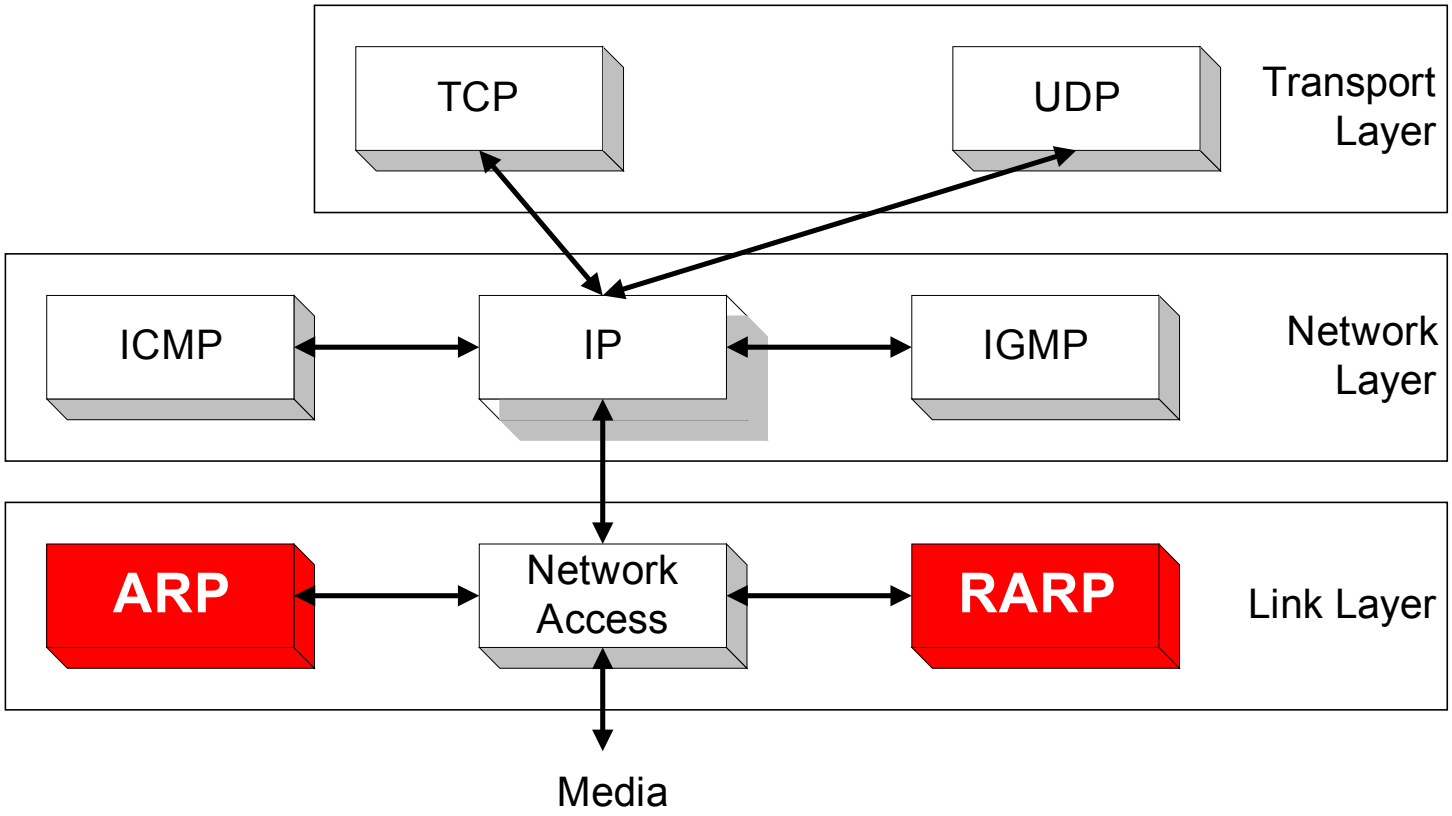
Differentiate v2 and 802.3: how?

- In order to allow some packets using Ethernet v2 framing and some packets using the original version of 802.3 framing to be used on the same Ethernet segment, EtherType values must be greater than 0x0600.
- That value was chosen because Ethernet 802.3 frames aren't supposed to exceed 1500 bytes (0x0600 = 1536 bytes > 1500 bytes).
- Thus if the field's value is greater than 0x0600, the frame must be an Ethernet v2 frame, with that field being a type field. If it's less than or equal to 0x0600, it must be an IEEE 802.3 frame, with that field being a length field.

ARP

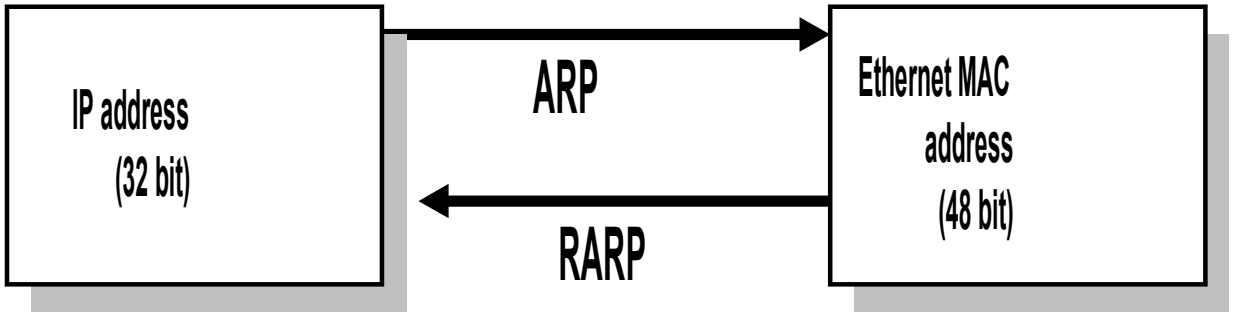
- Want to send a packet to a machine with the ip address X on the local network
- Must know the MAC address of this machine
- ARP: send an ethernet packet (frame) asking the MAC address for IP address X

Overview

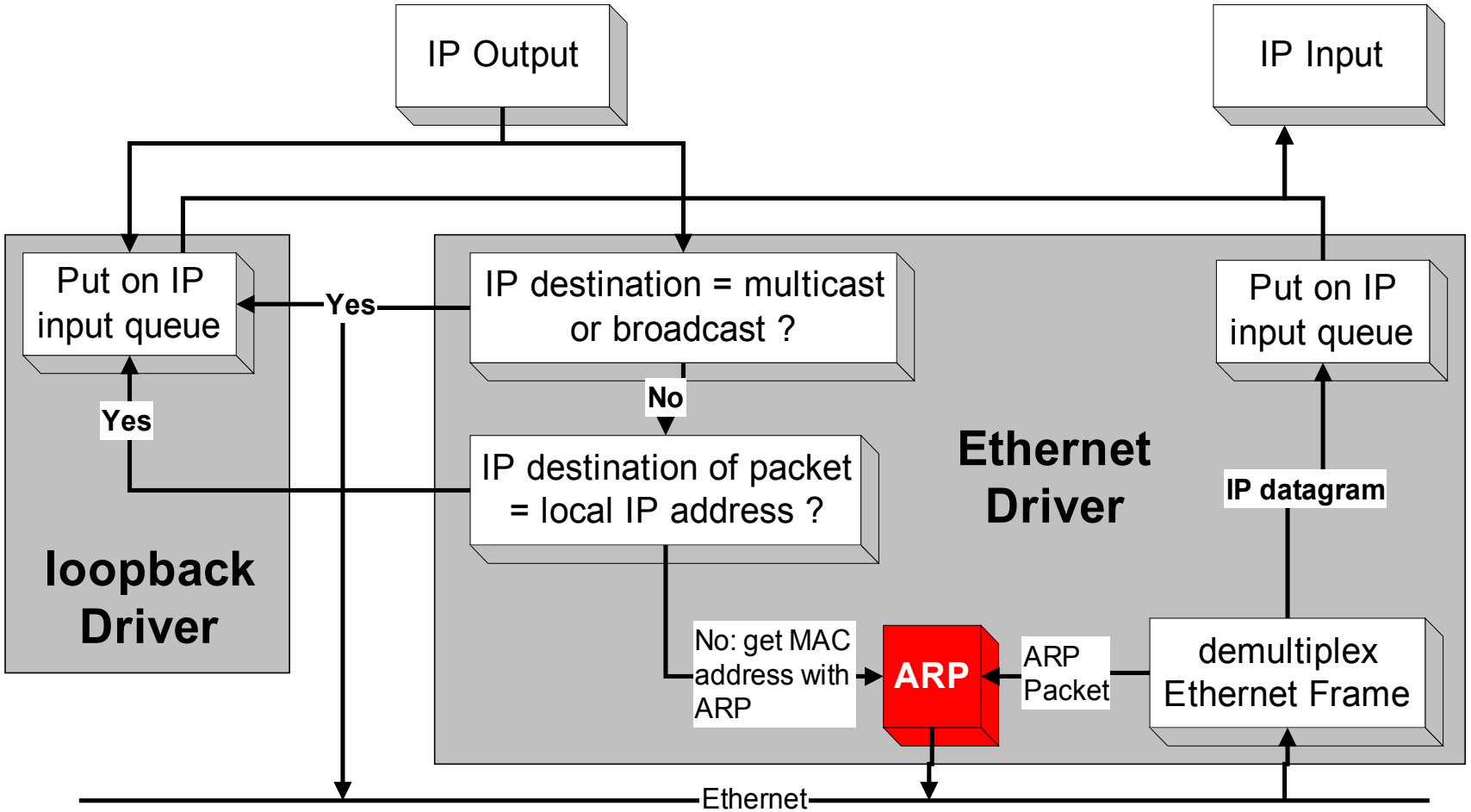

TCP
UDP
Transport Layer
ICMP
IP
IGMP
Network Layer
ARP
Network Access
RARP
Link Layer
Media

ARP and RARP

- Note:
  - The Internet is based on IP addresses
  - Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- The ARP and RARP protocols perform the translation between IP addresses and MAC layer addresses
- We will discuss ARP for broadcast LANs, particularly Ethernet LANs



Processing of IP packets by network device drivers

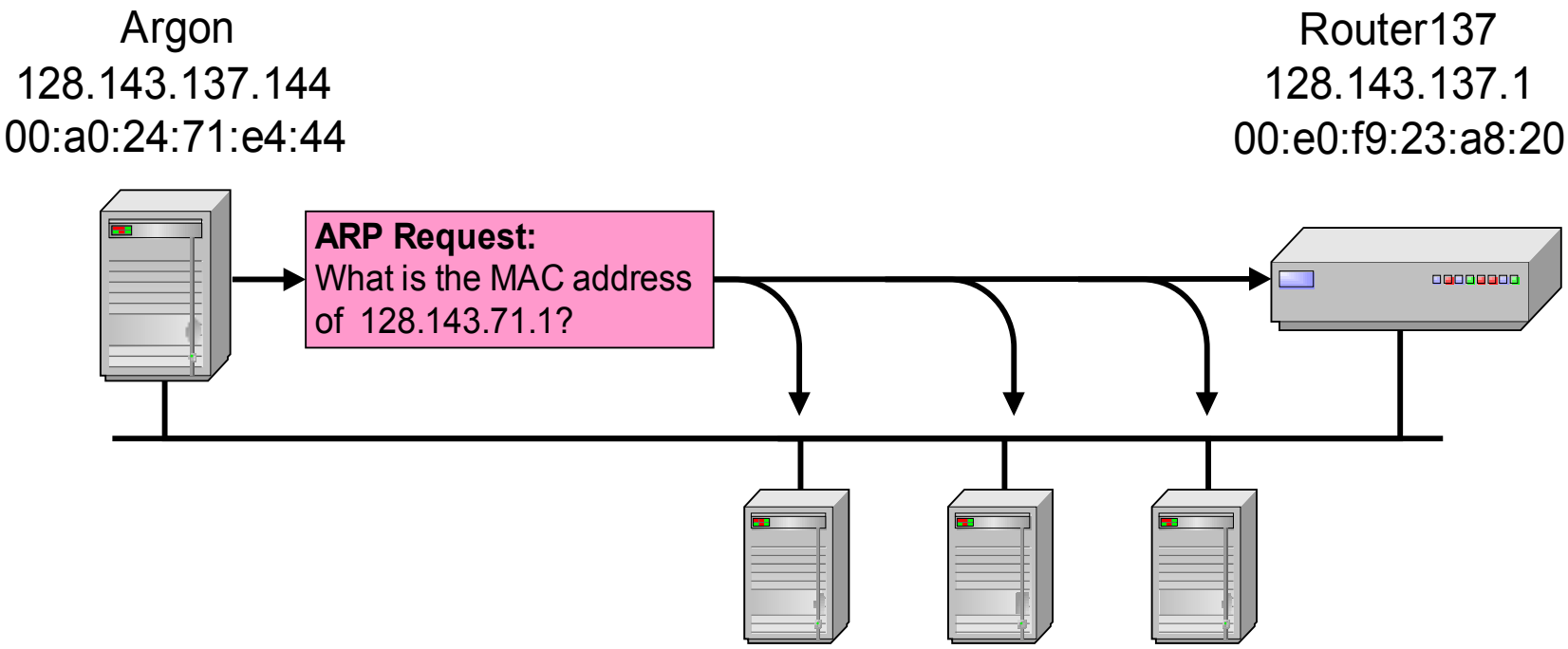


Address Translation with ARP

ARP Request:

Argon broadcasts an ARP request to all stations on the network:

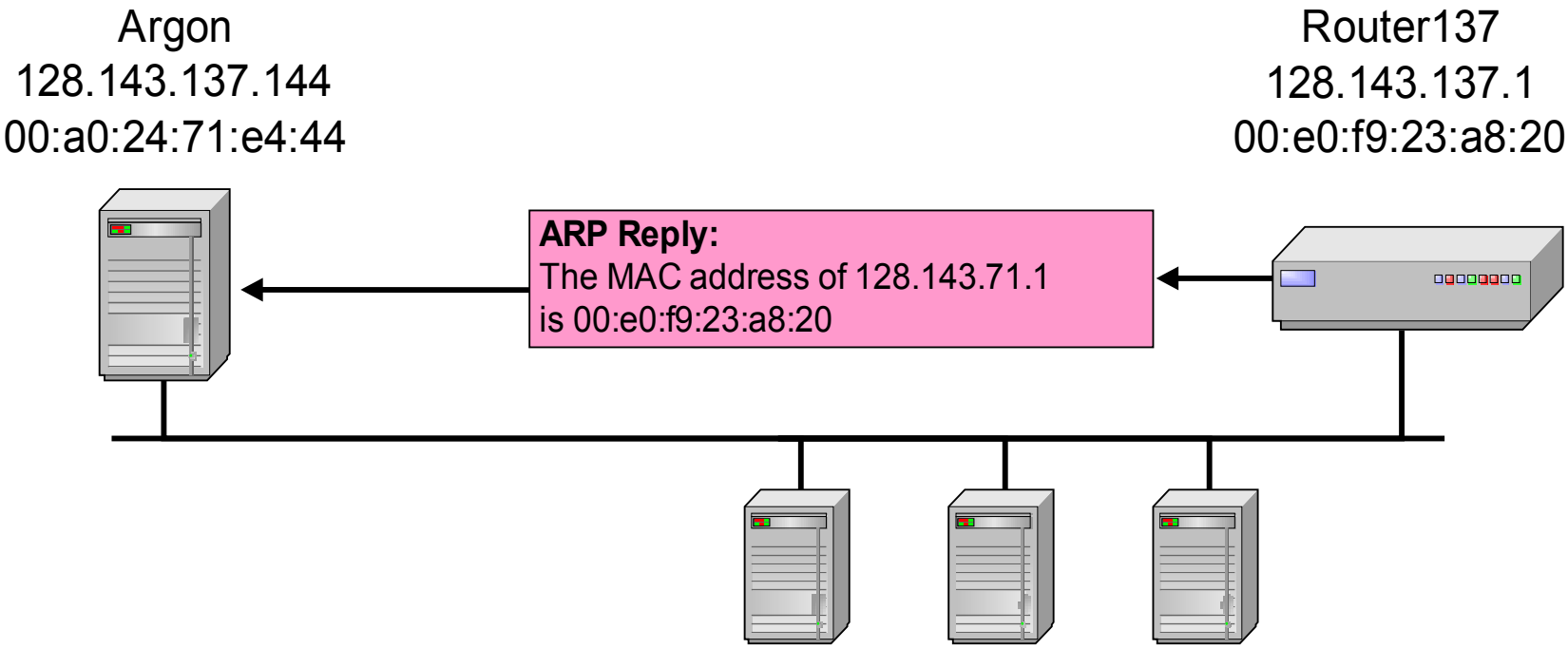
"What is the hardware address of Router137?"



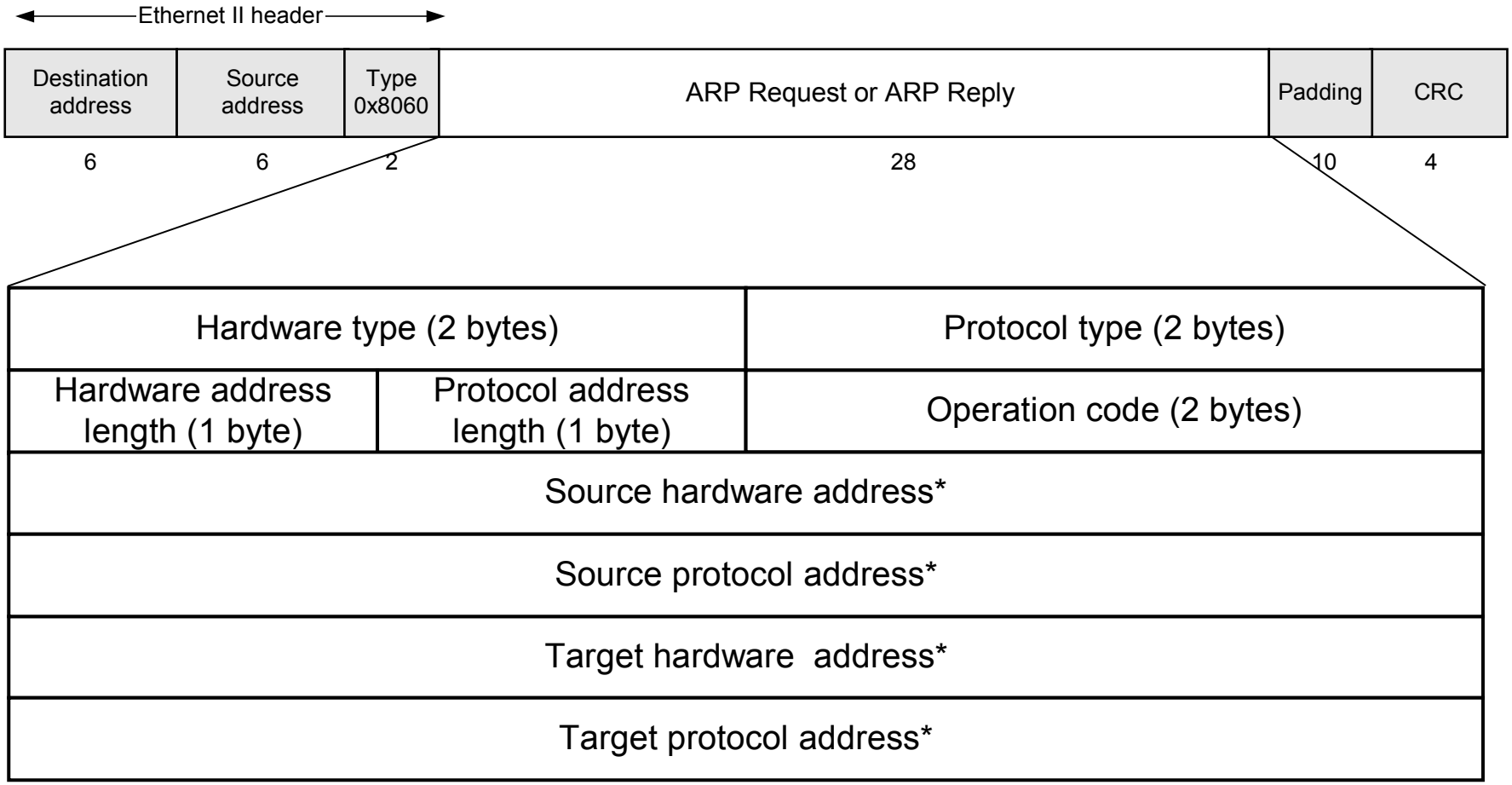
Address Translation with ARP

ARP Reply:

Router 137 responds with an ARP Reply which contains the hardware address



ARP Packet Format



* Note: The length of the address fields is determined by the corresponding address length fields

Example

- *ARP Request from Argon:*

  Source hardware address: 00:a0:24:71:e4:44
  Source protocol address: 128.143.137.144
  Target hardware address: 00:00:00:00:00:00
  Target protocol address: 128.143.137.1

- *ARP Reply from Router137:*

  Source hardware address: 00:e0:f9:23:a8:20
  Source protocol address: 128.143.137.1
  Target hardware address: 00:a0:24:71:e4:44
  Target protocol address: 128.143.137.144

ARP Cache

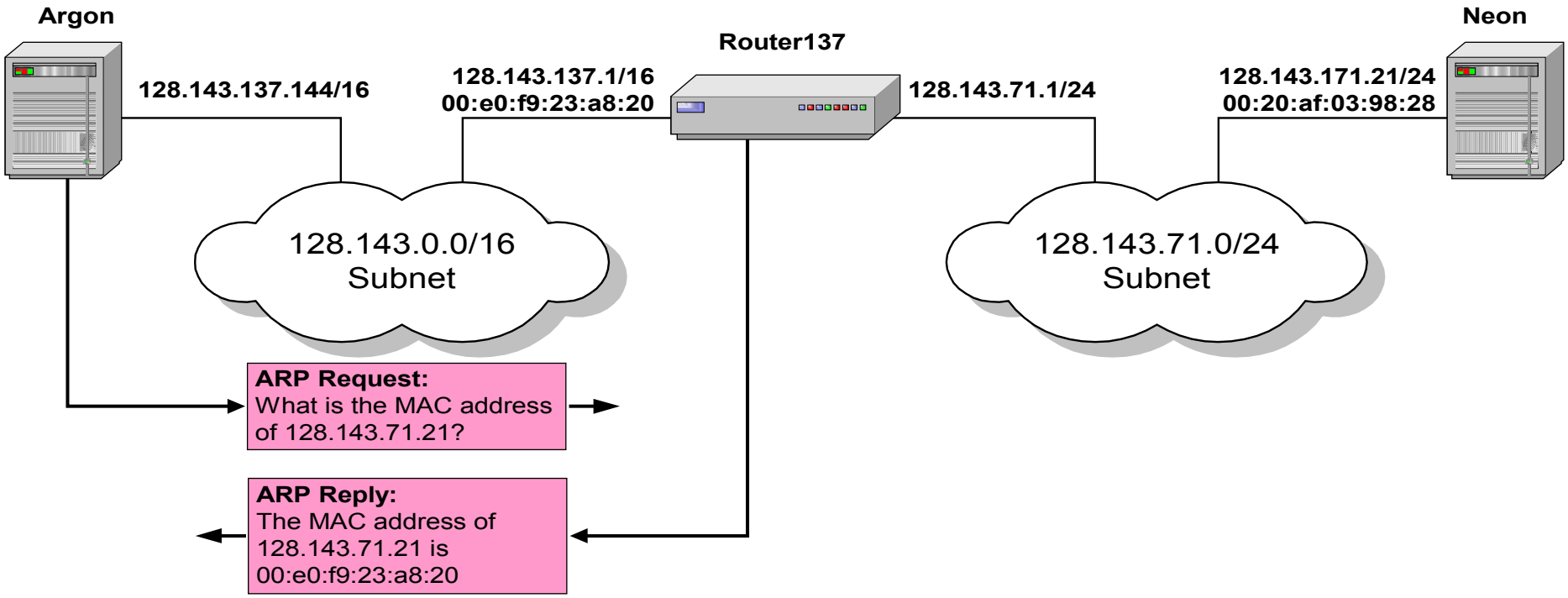
- Since sending an ARP request/reply for each IP datagram is inefficient, hosts maintain a cache (ARP Cache) of current entries. The entries expire after 20 minutes.

- Contents of the ARP Cache:

  (128.143.71.37) at 00:10:4B:C5:D1:15 [ether] on eth0
  (128.143.71.36) at 00:B0:D0:E1:17:D5 [ether] on eth0
  (128.143.71.35) at 00:B0:D0:DE:70:E6 [ether] on eth0
  (128.143.136.90) at 00:05:3C:06:27:35 [ether] on eth1
  (128.143.71.34) at 00:B0:D0:E1:17:DB [ether] on eth0
  (128.143.71.33) at 00:B0:D0:E1:17:DF [ether] on eth0

Proxy ARP

- **Proxy ARP:** Host or router responds to ARP Request that arrives from one of its connected networks for a host that is on another of its connected networks.



Things to know about ARP

- What happens if an ARP Request is made for a non-existing host?

  Several ARP requests are made with increasing time intervals between requests. Eventually, ARP gives up.

- On some systems (including Linux) a host periodically sends ARP Requests for all addresses listed in the ARP cache. This refreshes the ARP cache content, but also introduces traffic.

- Gratuitous ARP Requests: A host sends an ARP request for its own IP address:
  - Useful for detecting if an IP address has already been assigned.

Vulnerabilities of ARP

- Since ARP does not authenticate requests or replies, ARP Requests and Replies can be forged
- ARP is stateless: ARP Replies can be sent without a corresponding ARP Request
- According to the ARP protocol specification, a node receiving an ARP packet (Request or Reply) must update its local ARP cache with the information in the source fields, if the receiving node already has an entry for the IP address of the source in its ARP cache. (This applies for ARP Request packets and for ARP Reply packets)

Typical exploitation of these vulnerabilities:

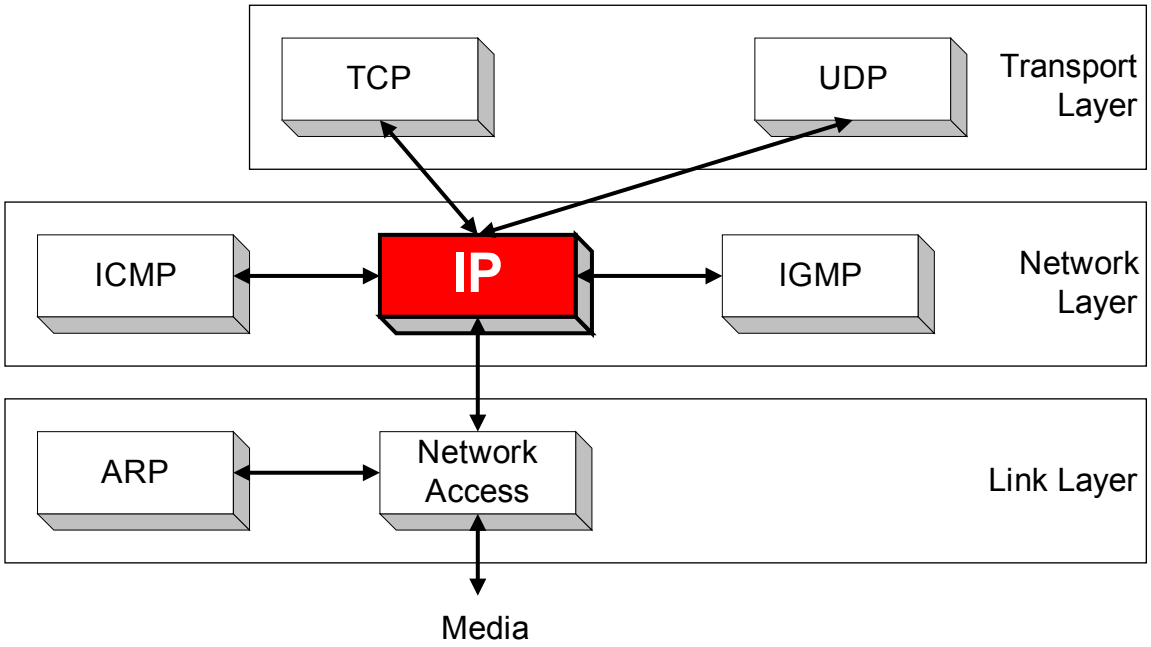
- A forged ARP Request or Reply can be used to update the ARP cache of a remote system with a forged entry (ARP Poisoning)
- This can be used to redirect IP traffic to other hosts

IP: internet protocol

- Ethernet frames do not reach routers outside our own network.
- Data is sent from my machine to far-away machines packaged into IP packets.
- IP packets are sent to the internet router (gateway) on our local network.
- The router then:
  - determines which router the packages have to be sent
  - sends the IP packages to the next router on the way
  - again packages our IP packets using ethernet or other standards

Orientation

- IP (Internet Protocol) is a Network Layer Protocol.

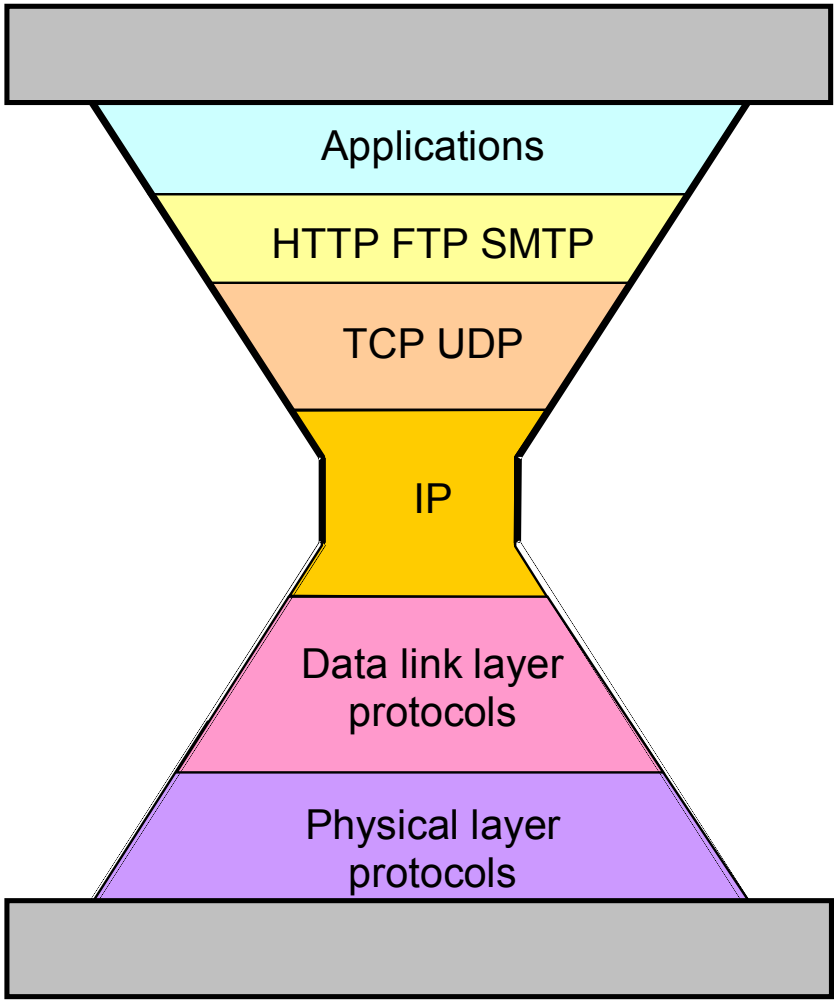


IP: The waist of the hourglass

- **IP is the waist of the hourglass of the Internet protocol architecture**

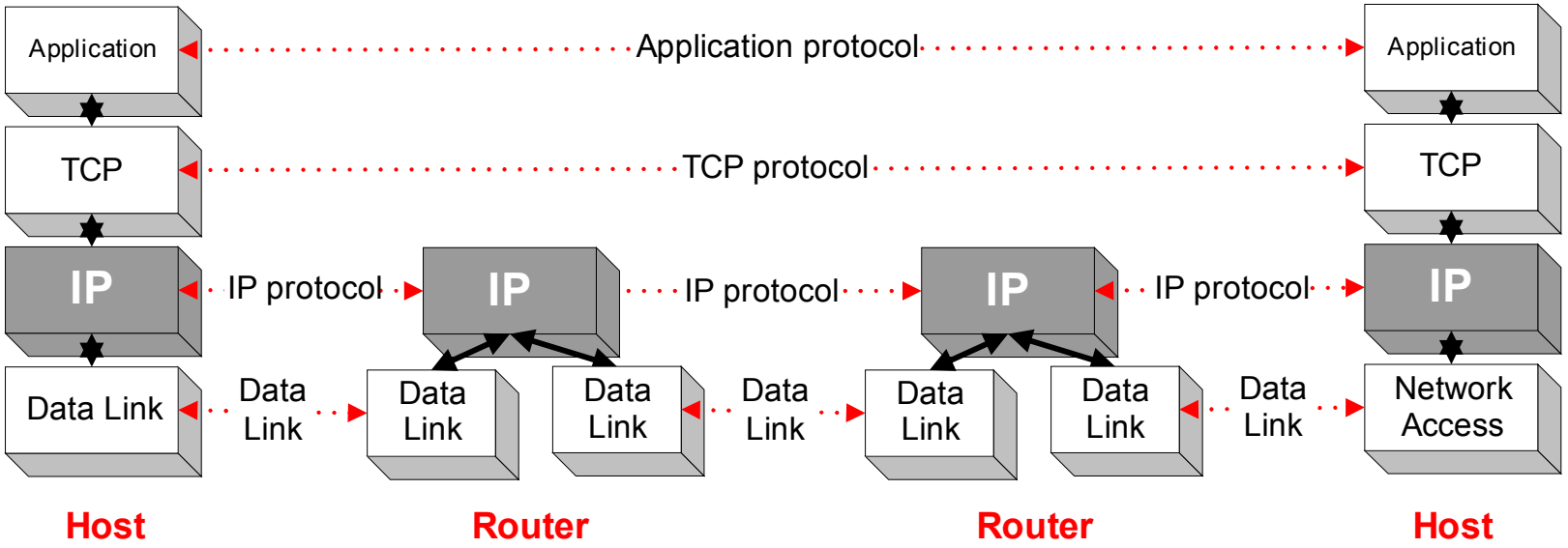
- Multiple higher-layer protocols
- Multiple lower-layer protocols

- Only one protocol at the network layer.



Application protocol

- IP is the highest layer protocol which is implemented at both routers and hosts

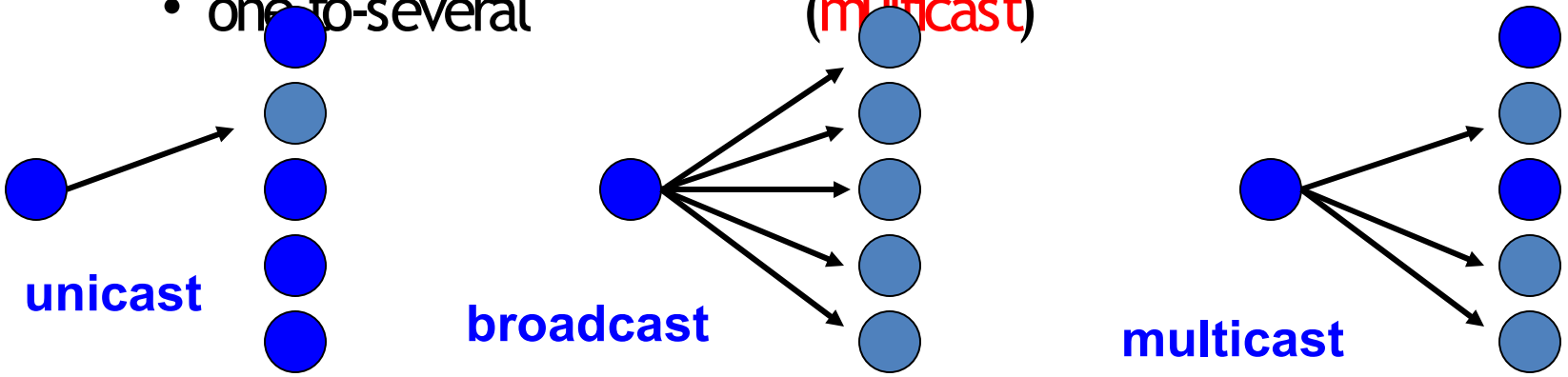


IP Service

- Delivery service of IP is minimal
- IP provide provides an unreliable connectionless best effort service (also called: "datagram service").
  - **Unreliable:** IP does not make an attempt to recover lost packets
  - **Connectionless:** Each packet ("datagram") is handled independently. IP is not aware that packets between hosts may be sent in a logical sequence
  - **Best effort:** IP does not make guarantees on the service (no throughput guarantee, no delay guarantee,...)
- Consequences:
  - Higher layer protocols have to deal with losses or with duplicate packets
  - Packets may be delivered out-of-sequence

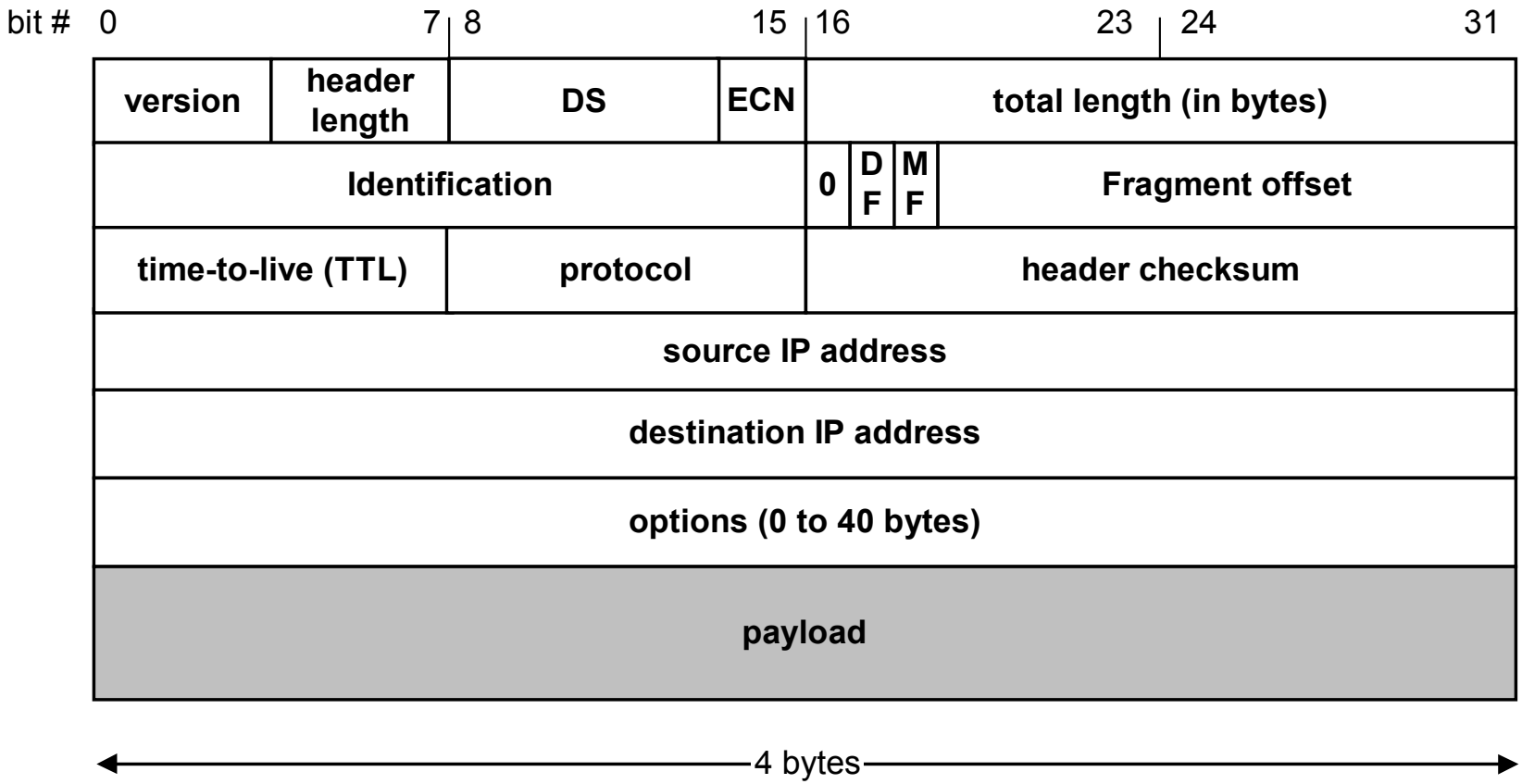
IP Service

- IP supports the following services:
  - one-to-one (unicast)
  - one-to-all (broadcast)
  - one-to-several (multicast)



- IP multicast also supports a many-to-many service.

IP Datagram Format



- 20 bytes ≤ Header Size < 2^4 x 4 bytes = 60 bytes
- 20 bytes ≤ Total Length < 2^{16} bytes = 65536 bytes

IP Datagram Format

- **Question:** In which order are the bytes of an IP datagram transmitted?
- **Answer:**
  - Transmission is row by row
  - For each row:
    1. First transmit bits 0-7
    2. Then transmit bits 8-15
    3. Then transmit bits 16-23
    4. Then transmit bits 24-31
- This is called **network byte** order or **big endian** byte ordering.

- **Note:** Many computers (incl. Intel processors) store 32-bit words in little endian format. Others (incl. Motorola processors) use big endian.

Big endian vs. small endian

- Conventions to store a multibyte work
- Example: a 4 byte Long Integer **`Byte3 Byte2 Byte1 Byte0`**

Little Endian

- Stores the low-order byte at the lowest address and the highest order byte in the highest address.

```
Base Address+0 Byte0
Base Address+1 Byte1
Base Address+2 Byte2
Base Address+3 Byte3
```

- Intel processors use this order

Big Endian

- Stores the high-order byte at the lowest address, and the low-order byte at the highest address.

```
Base Address+0 Byte3
Base Address+1 Byte2
Base Address+2 Byte1
Base Address+3 Byte0
```

Motorola processors use big endian.

Fields of the IP Header

- **Version (4 bits):** current version is 4, next version will be 6.
- **Header length (4 bits):** length of IP header, in multiples of 4 bytes
- **DS/ECN field (1 byte)**
  - This field was previously called as Type-of-Service (TOS) field. The role of this field has been re-defined, but is 'backwards compatible" to TOS interpretation
  - Differentiated Service (DS) (6 bits):
    - Used to specify service level (currently not supported in the Internet)
  - Explicit Congestion Notification (ECN) (2 bits):
    - New feedback mechanism used by TCP

Fields of the IP Header

- **Identification (16 bits):** Unique identification of a datagram from a host. Incremented whenever a datagram is transmitted

- **Flags (3 bits):**
  - First bit always set to 0
  - DF bit (Do not fragment)
  - MF bit (More fragments)

  Will be explained later→ Fragmentation

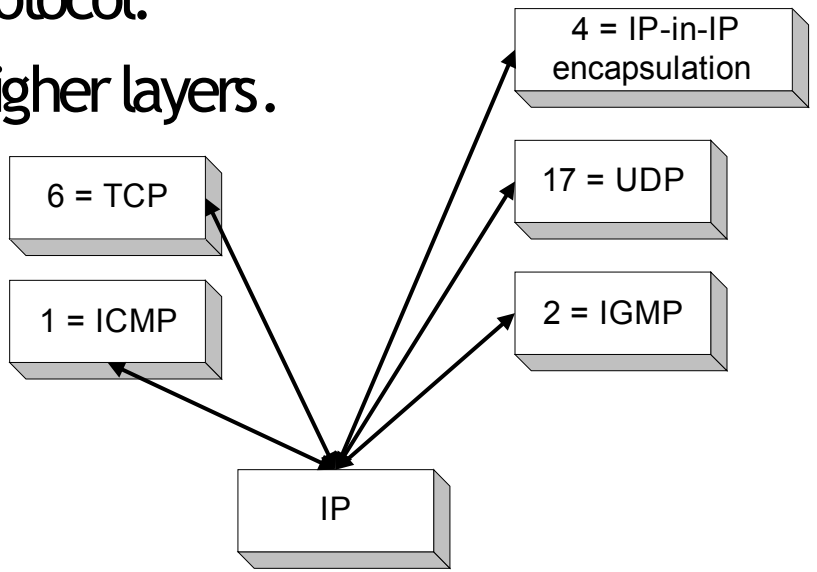
Fields of the IP Header

- **Time To Live (TTL) (1 byte):**
  - Specifies longest paths before datagram is dropped
  - Role of TTL field: Ensure that packet is eventually dropped when a routing loop occurs

  Used as follows:
  - Sender sets the value (e.g., 64)
  - Each router decrements the value by 1
  - When the value reaches 0, the datagram is dropped

Fields of the IP Header

- **Protocol (1 byte):**
  - Specifies the higher-layer protocol.
  - Used for demultiplexing to higher layers.



- **Header checksum (2 bytes):** A simple 16-bit long checksum which is computed for the header of the datagram.

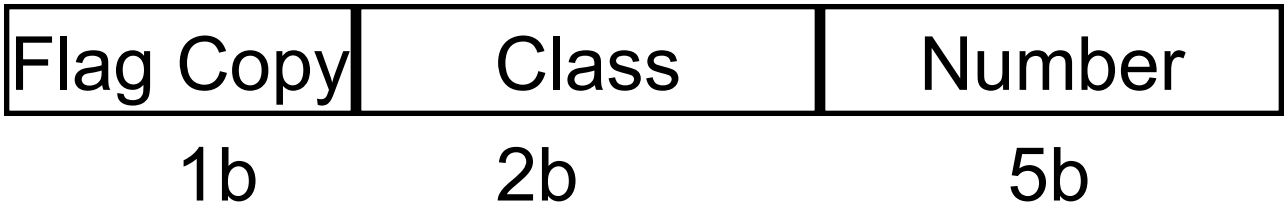
IP protocol numbers

- 1 byte: 256 numbers
  - 0 and 255 reserved
  - 103-254 unassigned
  - Most numbers are for exotic protocols
- Full list:
  - http://www.iana.org/assignments/protocol-numbers
  - http://en.wikipedia.org/wiki/List_of_IPv4_protocol_numbers

Fields of the IP Header

- **Options:**
  - Security restrictions
  - Record Route: each router that processes the packet adds its IP address to the header.
  - Timestamp: each router that processes the packet adds its IP address and time to the header.
  - (loose) Source Routing: specifies a list of routers that must be traversed.
  - (strict) Source Routing: specifies a list of the only routers that can be traversed.
- **Padding:** Padding bytes are added to ensure that header ends on a 4-byte boundary

IP Options Coding

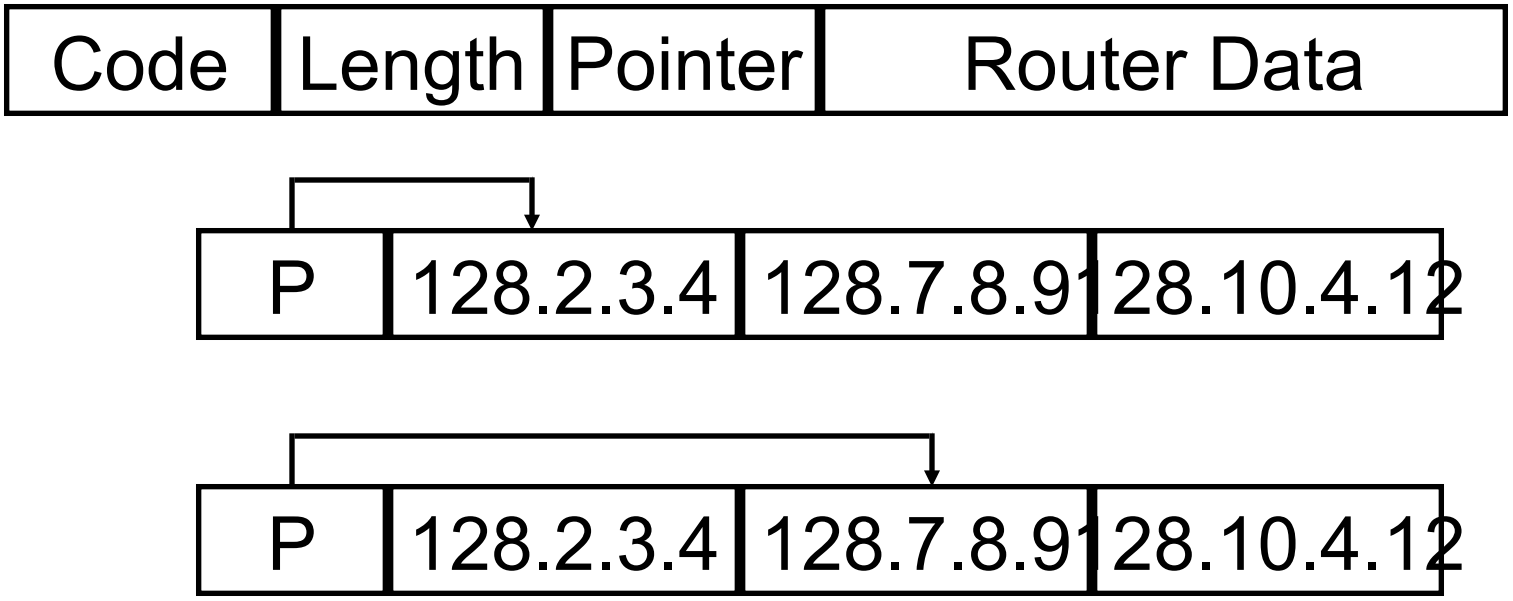


- Flag Copy: 0 = Copy the option only into the first fragment of a fragmented datagram
  1 = Copy into all fragments
- Class: 0 =User or control, 1=Reserved,
  2=Diagnostics, 3=reserved

IP Options

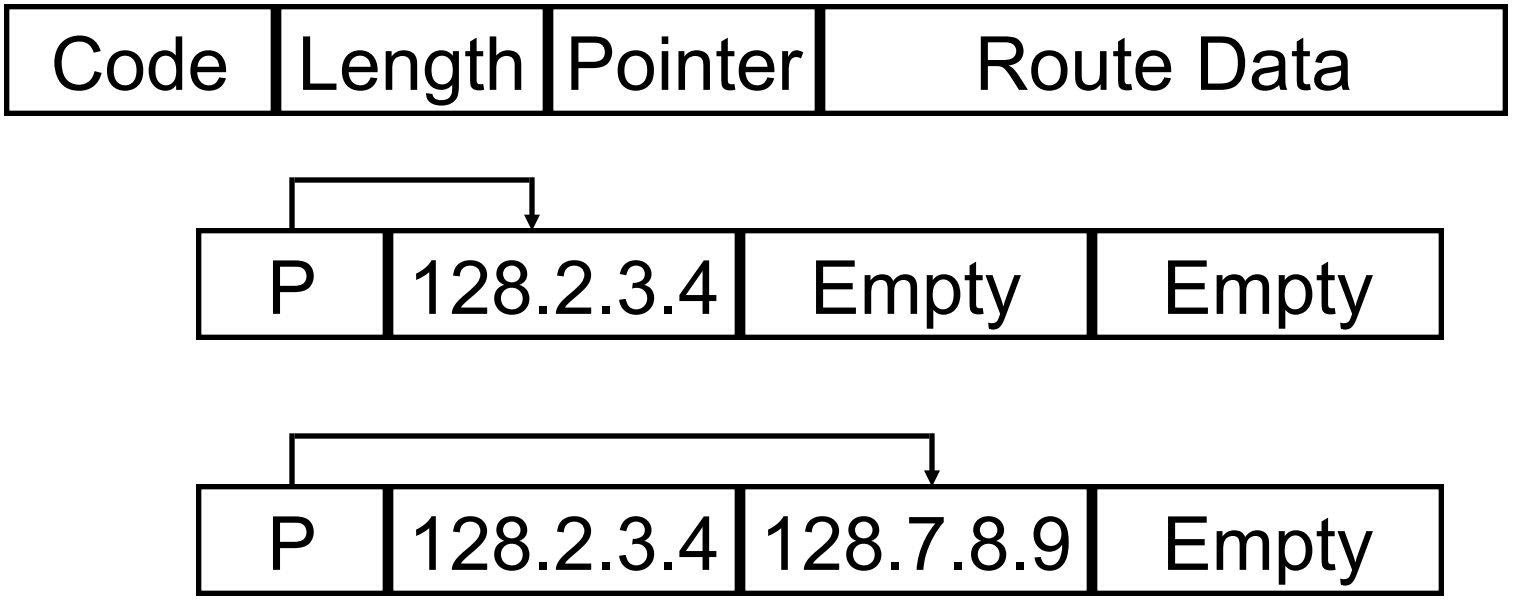
Class	Number	Length	Description
0	0	0	End of Options
0	1	0	No Op
0	2	11	Security
0	3	Var	Loose Source Routing
0	7	Var	Record Route
0	8	4	Stream ID (obsolete)
0	9	Var	Strict Source Routing
2	4	Var	Internet Time-Stamp

IP Source Routing



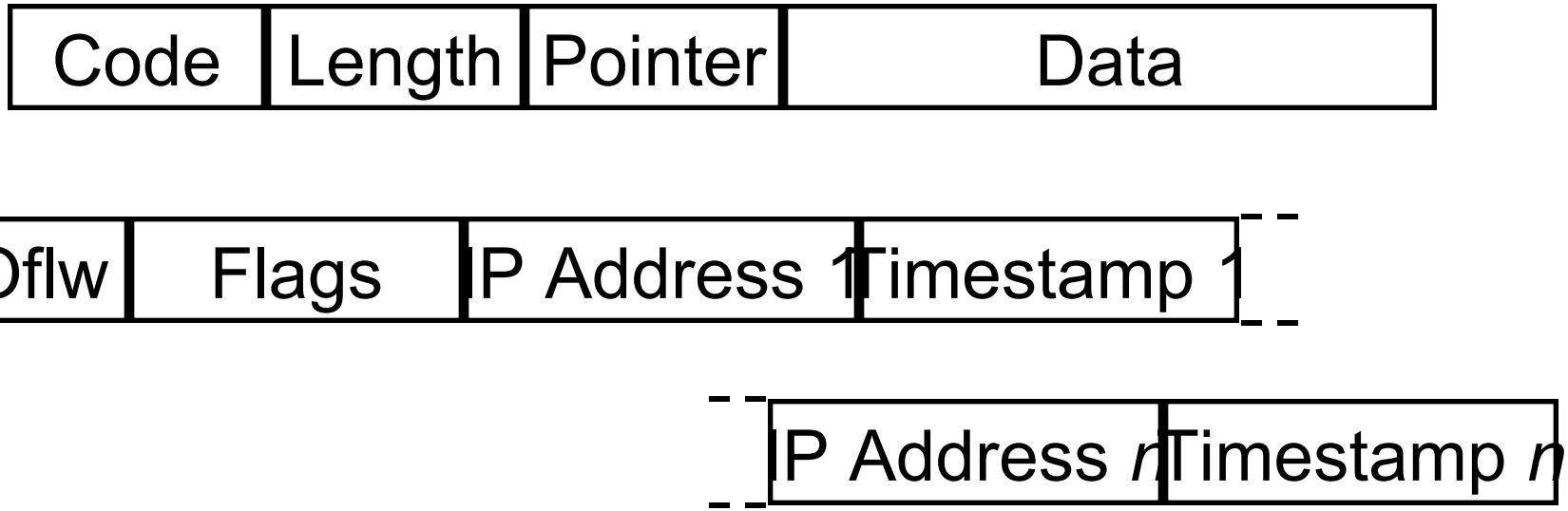
- Loose Source Routing (LSSR): Specify partial route list
- Strict Source Routing: Specify full route.

Route Recording



- Need to allow enough space to record IP addresses on route. Datagram size does not change as it goes through internet.

Timestamp Option



- Record timestamps along route
- Overflow (Oflw) counter incremented if out of space
- Flags: allows some further options for flexibility

Maximum Transmission Unit

- Maximum size of IP datagram is 65535, but the data link layer protocol generally imposes a limit that is much smaller

- Example:
  - Ethernet frames have a maximum payload of 1500 bytes
    → IP datagrams encapsulated in Ethernet frame cannot be longer than 1500 bytes

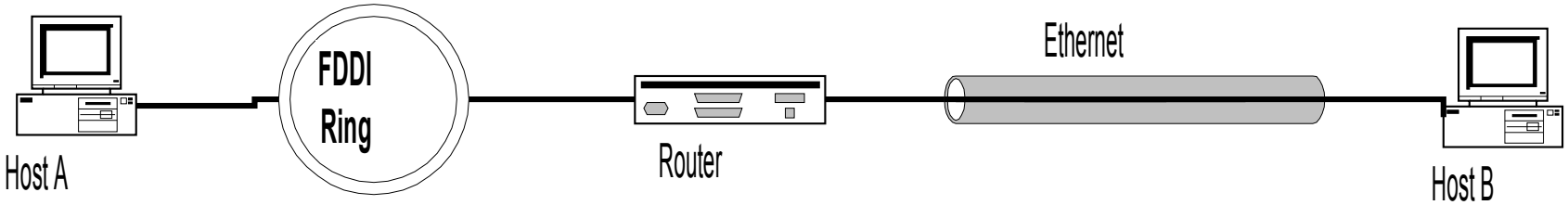
- The limit on the maximum IP datagram size, imposed by the data link protocol is called **maximum transmission unit (MTU)**

- MTUs for various data link protocols:

Ethernet:	1500	FDDI:	4352
802.3:	1492	ATM AAL5:	9180
802.5:	4464	PPP:	negotiated

IP Fragmentation

- What if the size of an IP datagram exceeds the MTU?
  IP datagram is fragmented into smaller units.
- What if the route contains networks with different MTUs?

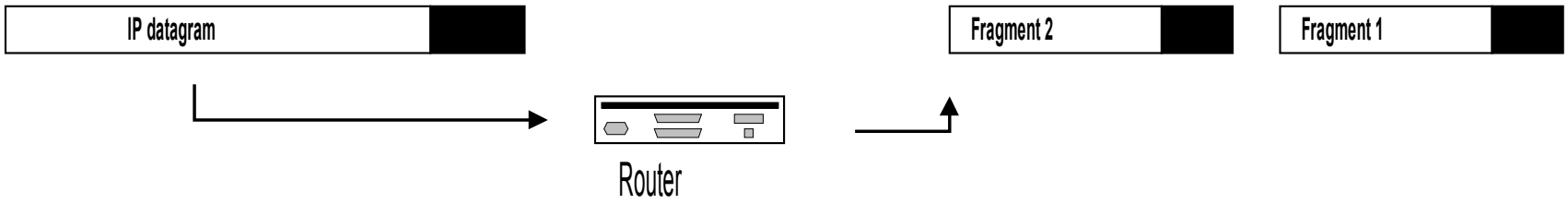


MTUs: FDDI: 4352 Ethernet: 1500

- **Fragmentation**:
  - IP router splits the datagram into several datagram
  - Fragments are reassembled at receiver

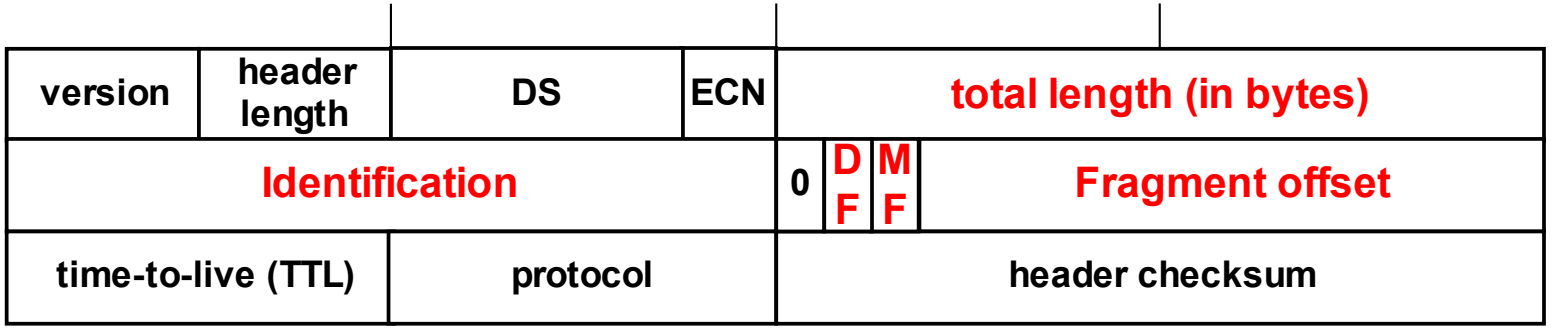
Where is Fragmentation done?

- Fragmentation can be done at the sender or at intermediate routers
- The same datagram can be fragmented several times.
- Reassembly of original datagram is only done at destination hosts !!



What's involved in Fragmentation?

- The following fields in the IP header are involved:



Identification — When a datagram is fragmented, the identification is the same in all fragments

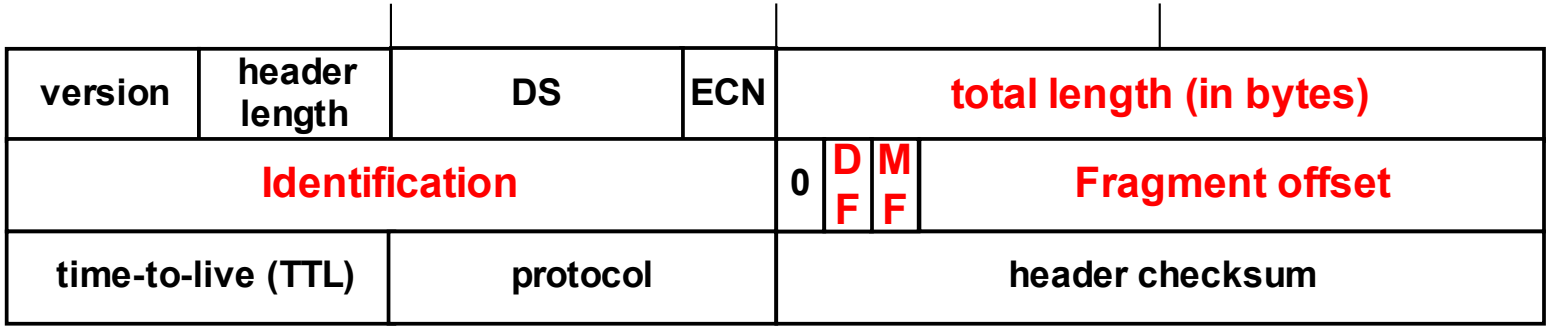
Flags

DF bit is set: Datagram cannot be fragmented and must be discarded if MTU is too small

MF bit set: This datagram is part of a fragment and an additional fragment follows this one

What's involved in Fragmentation?

- The following fields in the IP header are involved:



Fragment offset Offset of the payload of the current fragment in the original datagram

Total length Total length of the current fragment

Example of Fragmentation

- A datagram with size 2400 bytes must be fragmented according to an MTU limit of 1000 bytes

IP datagram	Fragment 3	Fragment 2	Fragment 1
Header length: 20	Header length: 20	Header length: 20	Header length: 20
Total length: 2400	Total length: 448	Total length: 996	Total length: 996
Identification: 0xa428	Identification: 0xa428	Identification: 0xa428	Identification: 0xa428
DF flag: 0	DF flag: 0	DF flag: 0	DF flag: 0
MF flag: 0	MF flag: 0	MF flag: 1	MF flag: 1
Fragment offset: 0	Fragment offset: 244	Fragment offset: 122	fragment offset: 0

MTU: 4000 MTU: 1000

Router

Determining the length of fragments

- To determine the size of the fragments we recall that, since there are only 13 bits available for the fragment offset, the offset is given as a multiple of eight bytes.
- As a result, the first and second fragment have a size of 996 bytes (and not 1000 bytes). This number is chosen since 976 is the largest number smaller than 1000-20= 980 that is divisible by eight.
- The payload for the first and second fragments is 976 bytes long, with bytes 0 through 975 of the original IP payload in the first fragment, and bytes 976 through 1951 in the second fragment.
- The payload of the third fragment has the remaining 428 bytes, from byte 1952 through 2379.
- With these considerations, we can determine the values of the fragment offset, which are 0, 976 /8 = 122, and 1952 /8 = 244, respectively, for the first, second and third fragment.